DOCUMENT RESUME

ED 282 714 SE 047 977

AUTHOR Embry, Lynn
TITLE Apple Treats. BASIC Classroom Computing Featuring Hands-On Activities and Pencil/Paper Fun.
REPORT NO ISBN-0-86653-326-5
PUB DATE 85
NOTE 100p.
AVAILABLE FROM Good Apple, Inc., P.O. Box 299, Carthage, IL 62321-0299 ($7.95 GA662).
PUB TYPE Guides - Classroom Use - Materials (For Learner) (051) -- Guides - Classroom Use - Guides (For Teachers) (052)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS Computer Graphics; *Computer Literacy; *Computer Science; Computer Uses in Education; Courseware; Educational Games; Elementary Education; *Elementary School Science; Glossaries; *Programing; Programing Languages; Puzzles; *Science Activities; Science Education; *Science Instruction
IDENTIFIERS *Apple (Computer); *BASIC Programing Language

ABSTRACT

Developed as an introduction to computer programming using an Apple microcomputer, this document presents teachers and students with: (1) essential vocabulary used in the BASIC language; (2) syntax; and (3) programming techniques. Simple hands-on activities designed to guide the learner through beginning BASIC programming skills and fundamental graphics are provided in the first half of the book. The second half contains activities, word puzzles and games, designed to evaluate student progress and reinforce the learning of computer programming. The document is designed to be used sequentially and many of the activities are printed in a reproducible format. Included in the back of the book are a summary of basic commands and keywords, a glossary, an answer key to the quizzes and activities, and an index. (TW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

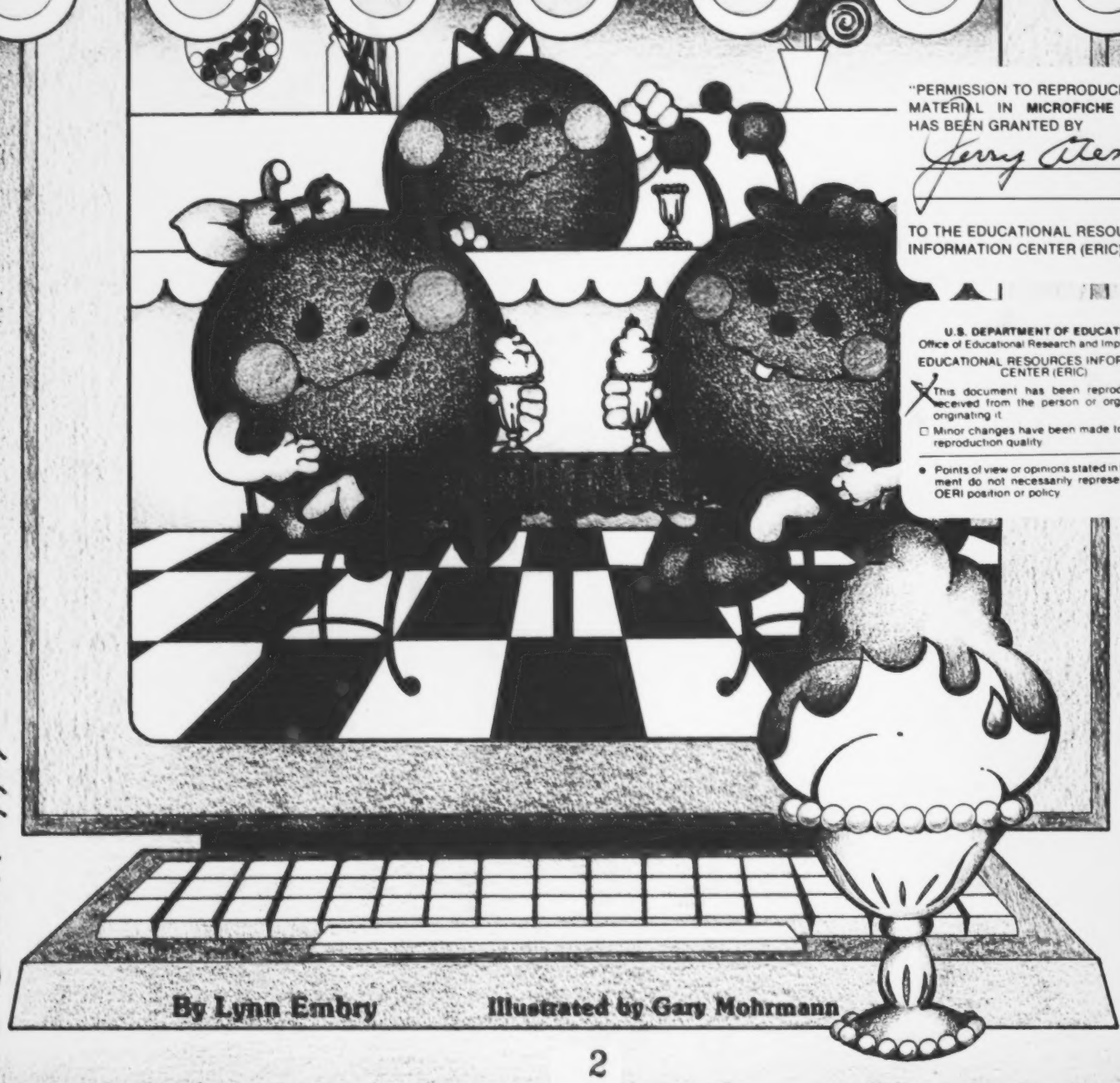
A Good Apple Activity Book
for Grades 4-12

Apple Treats

GOOD APPLE

ED287114

BASIC Classroom Computing Featuring Hands-On Activities and Pencil/Paper Fun



"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Jerry Aten

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

SE 044 773

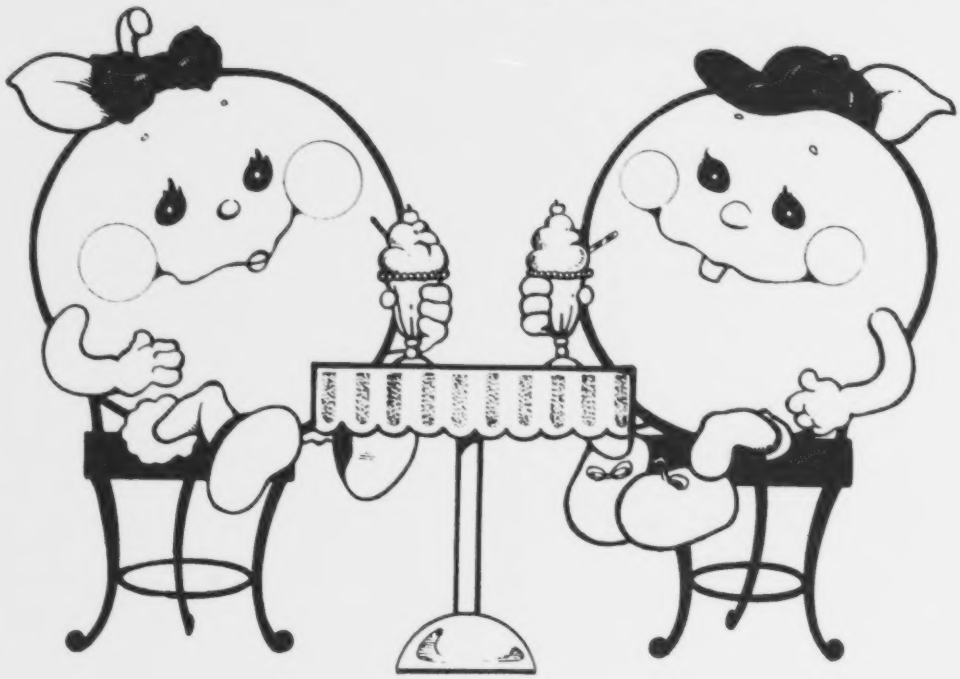
By Lynn Embry Illustrated by Gary Mohrmann

APPLE TREATS

BASIC Classroom Computing Featuring Hands-On Activities and Pencil/Paper Fun

by Lynn Embry

illustrated by Gary Mohrmann



Cover by Gary Mohrmann

Copyright © Good Apple, Inc., 1985

GOOD APPLE, INC.
BOX 299
CARTHAGE, IL 62321-0299

Copyright © Good Apple, Inc., 1985

ISBN No. 0-86653-326-5

Printing No. 987654321

GOOD APPLE, INC.
BOX 299
CARTHAGE, IL 62321-0299

The purchase of this book entitles the buyer to reproduce the student activity pages for classroom use only. Any other use requires written permission from Good Apple, Inc.

All rights reserved. Printed in the United States of America.

TABLE OF CONTENTS

Introduction iv
Introducing BASIC 1
Short BASIC Programs 2
Writing and LISTing a Program 3
Using LIST 4
On the RUN 5
Two Ways to PRINT 6
Multiple Statements 7
Let's Program 8
Tryout Time 9
BASIC Keywords 10
BASIC Commands 11
Information You Should Know 12
Syntax Error 13
Correcting Mistakes 14
Changing a Line 15
Deleting a Line 16
A BASIC Loop 17
Experimenting with GOTO 18
To Quote or Not to Quote 19
More About Quotation Marks 20
BASIC INPUT 21-22
Variables 23-24
LET 25
String Fever 26
REM 27
Using Concatenation 28-29
IF THEN 30-33
FOR NEXT 34
READ and DATA 35-36
Random 37-38
LEN 39-40
Spaced Out 41
TAB 42
HTAB and VTAB 43
Math Matters 44
Arithmetic Computations 45
Counting Programs 46
Using ◄, ►, and ◄► 47-48
Addition, Subtraction, and Multiplication 49
Long Division 50
Averaging Your Grades 51
Love Life Forecast 52
Winnings! 53
BASIC Bingo 54-55
Fortune Cookies 56
Drawing with the Keyboard 57
INVERSE 58
FLASH 59
Low-resolution Graphics 60-64
High-resolution Graphics 65-69
Which Is Correct? 70
Questions Often Asked About BASIC 71-72
Computer Quizzes 73-76
Computer Literacy Crossword 77
BASIC Word Search 78
BASIC Scrambler 79
Mystery Errors 80
Adventure Programs 81
Computer Word Links 82
Computer Literacy Word Search 83
Summary of Commands and Keywords 84
Glossary 85-87
Answer Key 88-90
Index 91-92

INTRODUCTION

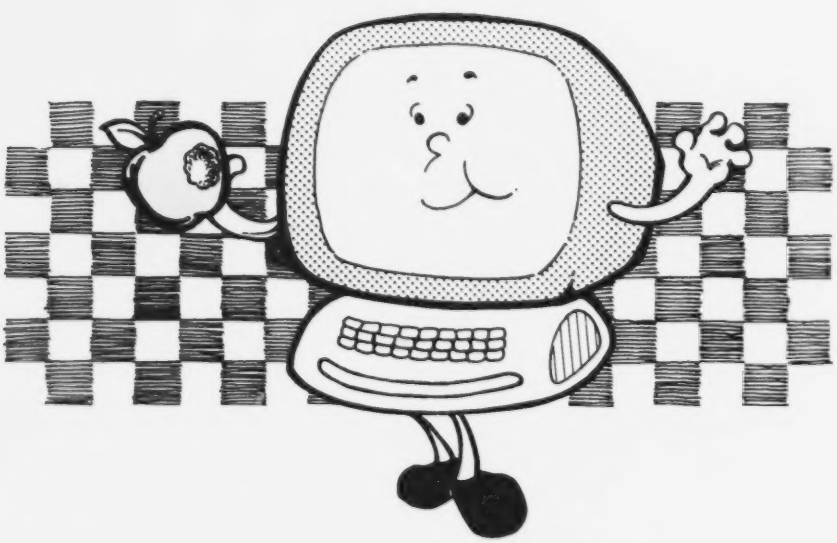
Computing with an *Apple is quite a treat! Its creative potential awaits your command.

Even if you have no experience with BASIC or computers, you and your students can create your own programs in a very short time. *Apple Treats* presents essential BASIC vocabulary, syntax, and programming techniques for the novice. Simple hands-on activities guide the learner through beginning BASIC programming skills and fundamental graphics. Checkup activities, word puzzles, and creative ideas are included to evaluate progress and reinforce learning.

Apple Treats is intended to be an introduction to BASIC, but its scope is advanced enough to start you on your way to developing your own computer programs. It is recommended that the learner start at the beginning of the book and work through the lessons in the order of presentation. As skills are acquired and confidence gained, the learner is encouraged to experiment with ideas of his own to fully enjoy the treat of creative computing.

Happy programming!

*Apple and Applesoft are registered trademarks of Apple Computer, Inc.



INTRODUCING BASIC

The computer language you are about to learn is called BASIC. BASIC means "Beginner's All-purpose Symbolic Instruction Code." BASIC has some important vocabulary words, a grammatical structure, and special words that must be followed in order to communicate with the computer.

The computer translates BASIC into a machine code. It is not necessary to understand the machine code because a BASIC compiler does the translating. However, it is necessary to understand BASIC language in order to instruct the computer to perform a task.

WHAT IS A PROGRAM?

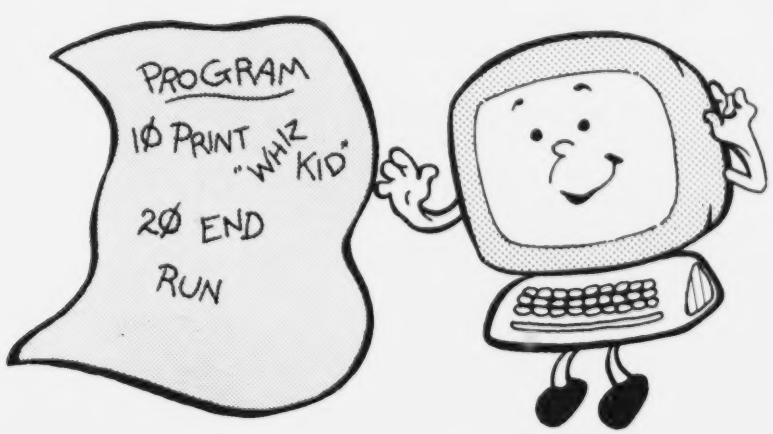
A program is a list of numbered instructions for the computer to follow. A program may be simple or complicated.

Here is a simple program in BASIC to try with your computer. Type each line exactly as you see it. Be sure to use capital letters. On most keyboards you must press shift and the key containing the quotation mark to get the quotation mark to print. You must press the RETURN key after each line you type.

```
1Ø PRINT "WHIZ KID"
2Ø END
```

Now type RUN and press RETURN. What happened? Record your answer here.

If you followed the directions, the computer should have printed WHIZ KID. If it did not work for you, try it again. Be sure you type it exactly as you see it. Leave a space after each line number and between each word, and use quotation marks where they are indicated.



SHORT BASIC PROGRAMS

Try this program. First, notice the line number in the first statement. The zero in the ten has a slash mark through it. That is so you will not confuse it with the letter "O" when you type. Be sure to type a zero instead of the letter "O" when using numerals. Otherwise, you will receive a syntax error message. (Syntax error messages will be explained later.)

```
1Ø PRINT "THIS IS GREAT."
2Ø END
```

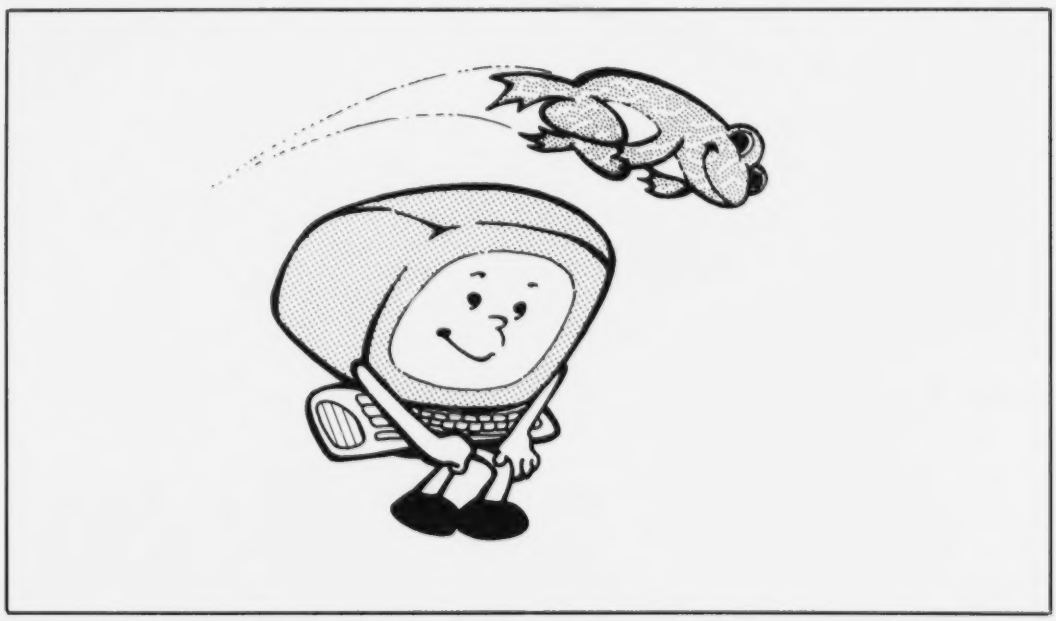
Now type RUN and press RETURN.

What did the sheep say when he typed his first program? RUN the program and see!

```
1Ø PRINT "NOT"
2Ø PRINT "BAAAD"
3Ø END
RUN
```

A program in BASIC uses line numbers to direct the computer in carrying out its instructions. The computer goes to the lowest line number first and follows its orders. Then it goes to the next higher number and carries out its orders. It continues in numerical order unless it is directed to a different line number.

Q: What did the frog say when he wrote his first program?
A: "I'm so hoppy!"



WRITING AND LISTING A PROGRAM

Here is a simple program to type. Type it exactly as you see it.

```
10 PRINT "SPECIAL"
20 PRINT "PERSON"
30 END
```

Type RUN and press RETURN to get the computer to follow your instructions. Now type LIST and press RETURN. Your program instructions should appear on the screen.

Look at the program instructions. Each line tells the computer something to do. Line 10 instructs the computer to print the word SPECIAL. Line 20 instructs the computer to print the word PERSON. Line 30 ends the program. Actually, the program will run without line 30, but it is considered good practice to designate the end of a program.

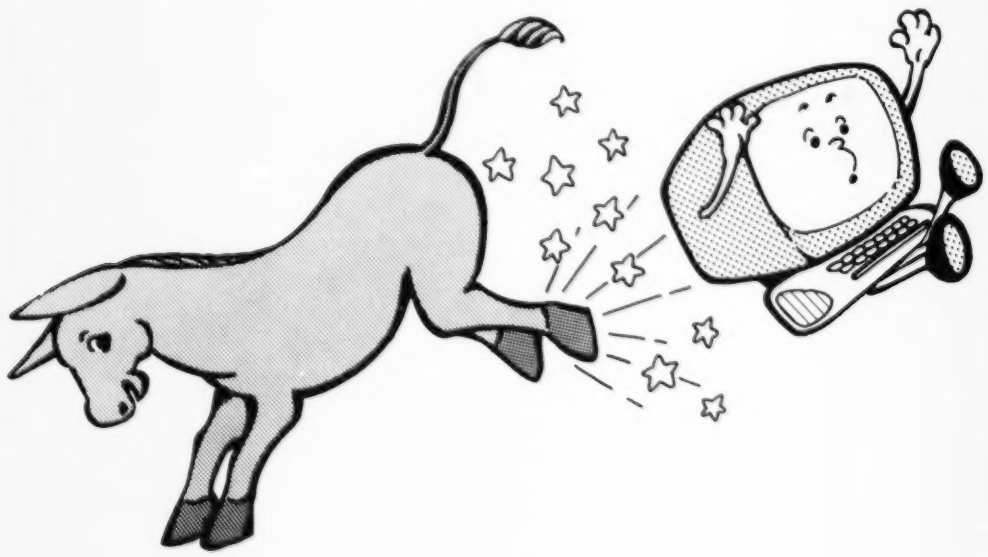
SUMMARY:

1. LIST displays the program instructions. It prints the line numbers and statements without carrying them out. The line numbers are listed in chronological order.

2. RUN carries out the program instructions. It is the command that causes the program to work.

3. LIST and RUN are BASIC commands. They are called commands because they tell the computer to do something.

Q: What did the mule say when he ran his first program?
A: "I get a bit of a kick out of this!"



USING LIST

You can use the LIST command in several ways:

1. You already know if you type LIST and press RETURN, the program instructions will appear on the screen.

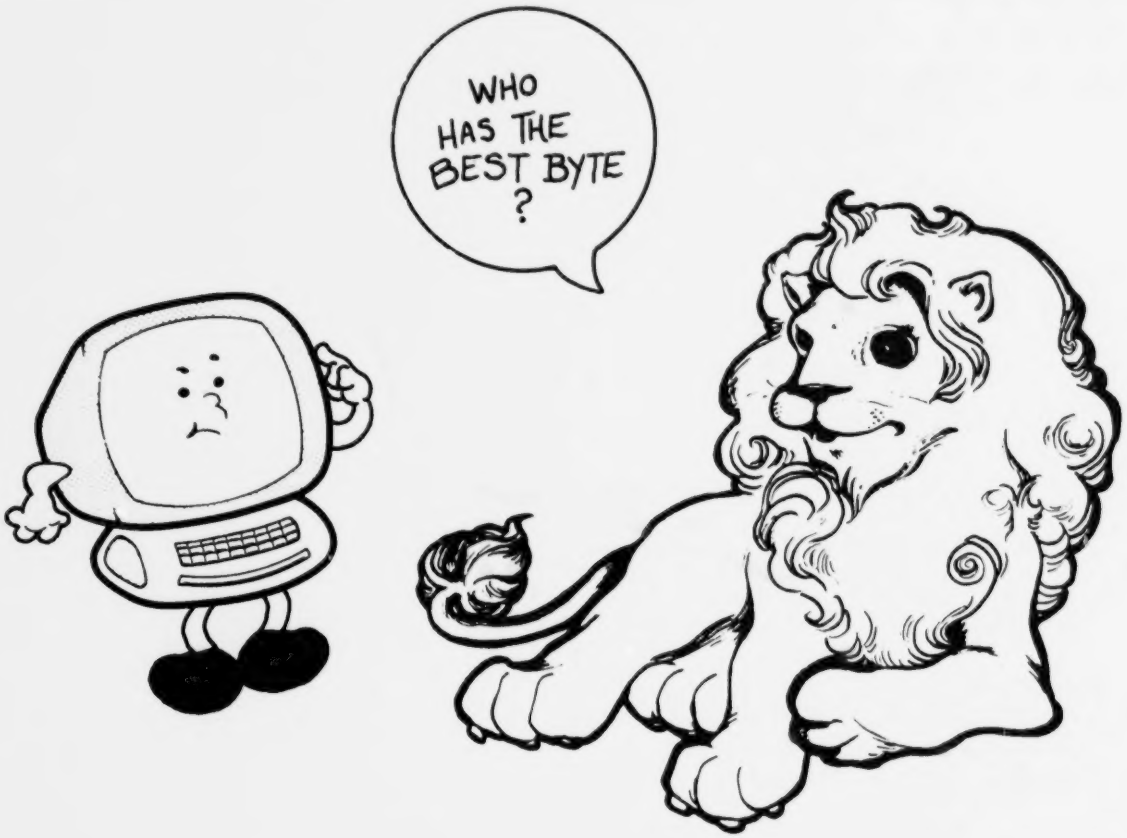
2. You can also instruct the computer to display one particular program line. If you type LIST 6Ø, the monitor will display line 6Ø of your program.

3. Programs can be listed from the beginning to a specific line number. Type LIST-15Ø and press RETURN. The program will be listed from the beginning through line 15Ø.

4. Another way to use LIST is to type a line number followed by a dash. LIST 15Ø- will list instructions beginning with line 15Ø and continuing to the end of the program.

5. What will happen if you type LIST 1Ø-15Ø and press RETURN?

Q: What did the lion ask the computer?
A: "Who has the best byte?"



ON THE RUN

RUN is the command used to tell the computer to work your program. It is not a statement, so it does not have a line number. Always press RETURN after typing RUN. You will not be reminded to press RETURN after this page.

Type and RUN this program:

```
10 PRINT "HELLO THERE."
20 END
RUN (Press RETURN.)
```

What happened? _____

We measure a year by the length of time it takes Earth to revolve around the sun. Suppose you want to know your age on Mercury. Here is a program to compute your age in Mercury's years. Type and RUN the program.

```
10 PRINT "WHAT IS YOUR NAME?";N$
20 INPUT N$
30 PRINT "HOW MANY YEARS OLD ARE YOU?"
40 INPUT A
50 X = A*4  (space)                                    (space)
60 PRINT N$;",↓YOU ARE A LITTLE MORE THAN↓";↓X;
   " YEARS OLD IN MERCURY'S YEARS."
70 END  (space)
RUN (Press RETURN.)
```

Q: What did the lion say when asked if he liked computers?
A: "Sure. Computers are my mane interest."

TWO WAYS TO PRINT

You have learned to use PRINT in a program statement. You can also use a question mark (?) instead of typing the word PRINT. The computer responds in the same way. In fact, the Apple and some other computers even display the word PRINT when the program is LISTed.

Try this:

```
10 ? "ROSES ARE RED;"
20 ? "VIOLETS ARE BLUE."
30 ? "I PRINTED WITH A QUESTION MARK."
40 ? "HOW ABOUT YOU?"
50 END
```

RUN the program. Then LIST the program instructions. What does the computer display?

_____

Now try this:

```
10 PRINT "THIS"
20 PRINT "IS"
30 PRINT "A"
40 PRINT "PROGRAM."
50 END
RUN
```

Compare the program above with this one.

```
10 ? "THIS"
20 ? "IS"
30 ? "A"
40 ? "PROGRAM."
50 END
RUN
```

What did you discover when you compared the two programs?

_____

Type and RUN this program.

```
10 PRINT "THIS IS A PROGRAM."
20 END
RUN
```

How is this version different?_____

_____

MULTIPLE STATEMENTS

More than one statement can be entered for a program line if each statement is separated by a colon (:). This technique may be helpful for saving computer memory space. It also speeds the execution of the program.

The maximum number of characters or keystrokes that may be entered for a line number is 239.

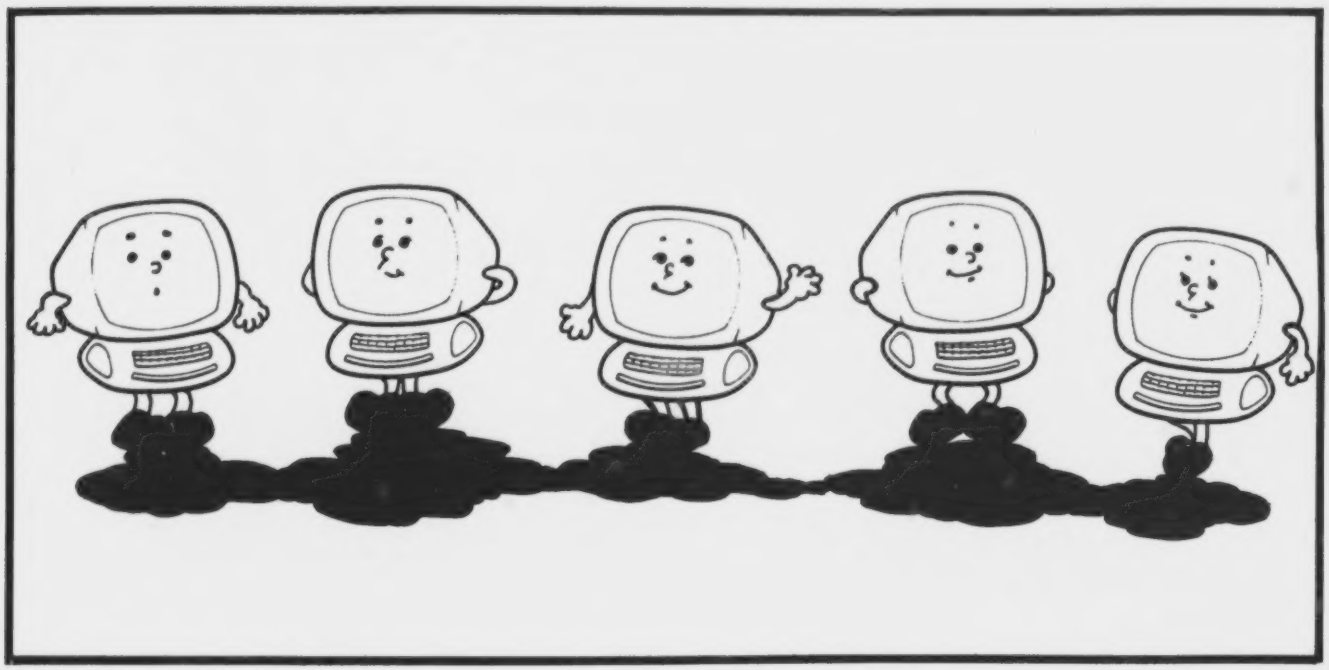
EXAMPLE:

```
10 PRINT "WHAT IS YOUR NAME?":INPUT N$:PRINT:PRINT "NICE TO MEET
   YOU ";N$"."
       (space)
```

Here is a program containing multiple statements. You may wish to try it.

```
10 PRINT "THINK OF A NUMBER.":PRINT "FOLLOW MY DIRECTIONS.":
   PRINT "I WILL GUESS YOUR NUMBER."
20 PRINT:PRINT "DON'T TELL ME YOUR NUMBER.":PRINT:PRINT "ADD 5
   TO THE NUMBER."                                    (space)
30 PRINT:PRINT "SUBTRACT 2.":PRINT:PRINT "MULTIPLY BY 3."
40 PRINT :PRINT "ADD 5.":PRINT "WHAT DID YOU GET?"
50 PRINT:INPUT A:X = (A - 5)/3 + 2 - 5
60 PRINT "YOUR NUMBER IS ";X".":END
RUN                        (space)
```

ON YOUR OWN: Write a program that contains at least five multiple statements. RUN your program to be sure it does what you anticipate.



LET'S PROGRAM!

Study the program below. It prints out a name and address. Notice that only the words and numbers inside quotation marks are displayed when the program is RUN. Try it and see.

```
10 PRINT "GREG NELSON"
20 PRINT "204 ELM STREET"
30 PRINT "CAIRO, GEORGIA 31728"
40 END
RUN
```

GREG NELSON
204 ELM STREET
CAIRO, GEORGIA 31728

← Does your screen display look like this?

Now type a program that prints your name and address. RUN the program.

QUESTIONS:

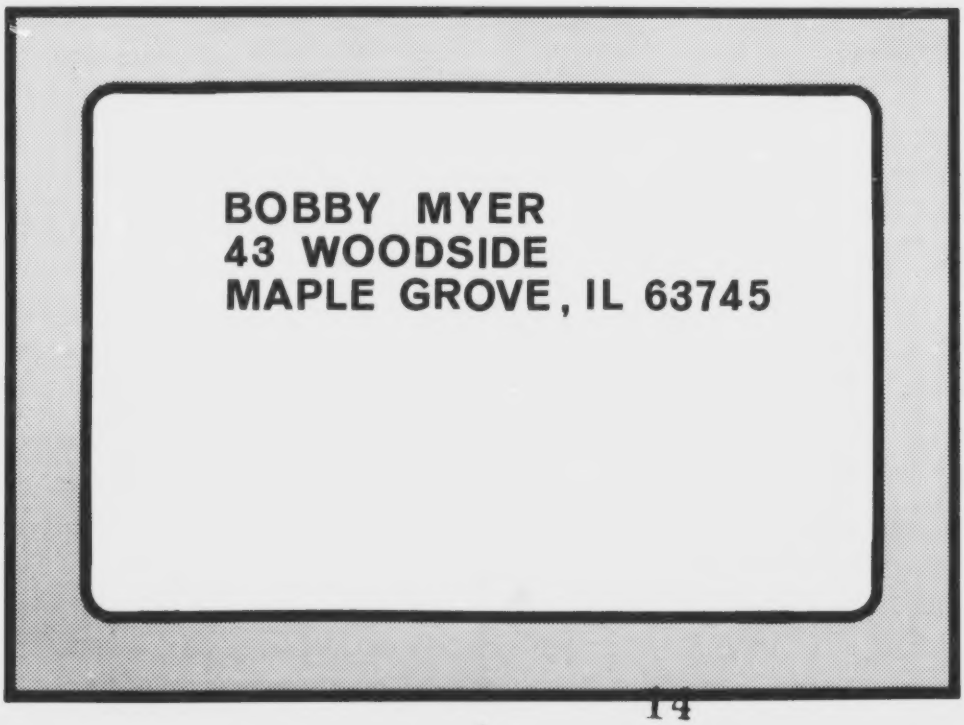
1. Why are quotation marks necessary in lines 10, 20, and 30 of the program?

   _____

2. What will happen if the quotation marks are left out?

   _____

ON YOUR OWN: Write a program that lists your friends' names and telephone numbers.



TRYOUT TIME

Write a BASIC program that will print an identification card. Include your name, address, telephone number, parents' names, and the name and telephone number of a person to contact in case of emergency.

Write your program in the space below. Then type it on the keyboard and RUN it.

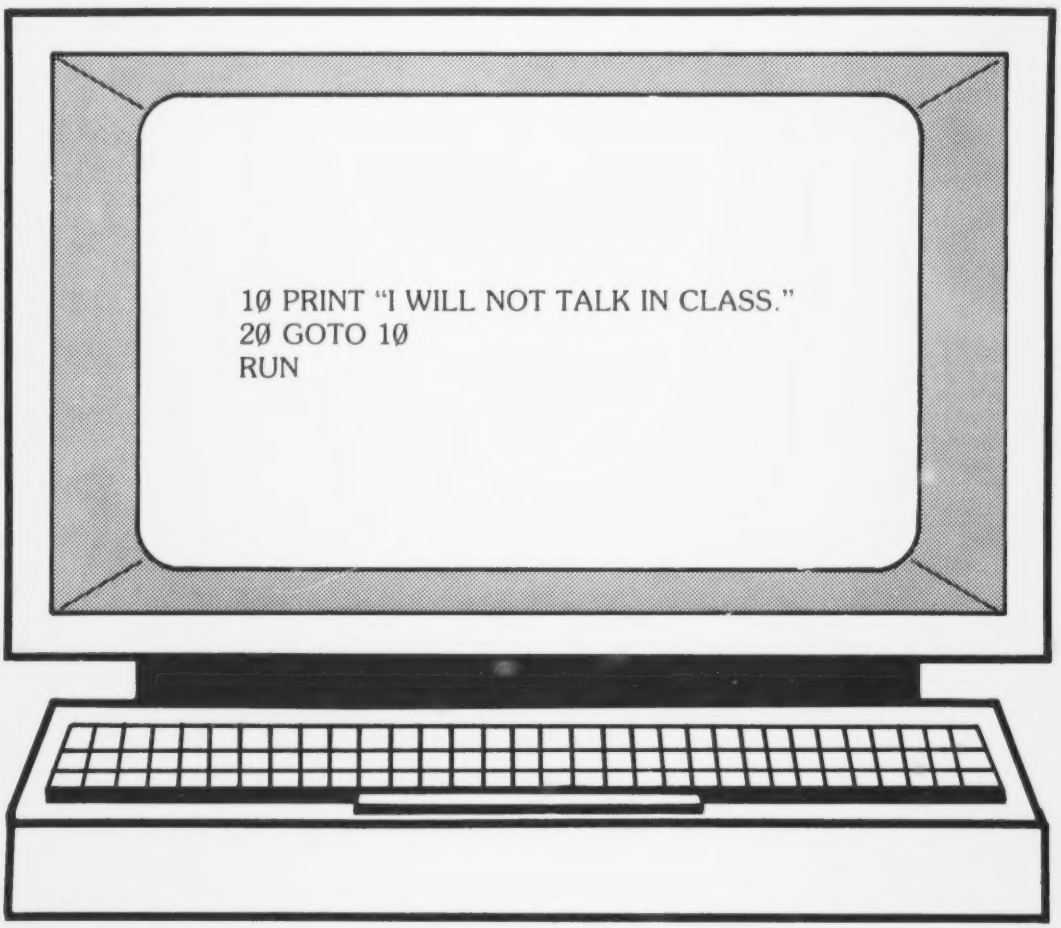
HINT: Use a different line number for each piece of information.

BASIC KEYWORDS

Keywords are used to make statements, and statements are used to make programs.

Below are some BASIC keywords. Don't worry if you don't know what they all mean. Soon you will be using them in your programs.

PRINT	STOP
LET	FOR . . . (STEP)
INPUT	NEXT
GOTO	END
IF . . . THEN	REM



BASIC COMMANDS

Here are two BASIC commands that you have used. Commands tell the computer to do something. Commands do not use line numbers.

RUN

LIST

Do you remember what RUN and LIST do? When you type RUN and press RETURN, the computer carries out the program. When you type LIST and press RETURN, the computer shows the program instructions on the screen.

Here are some other useful commands.

NEW: clears a program from memory. NEW is used when the user is ready to enter a different program. If the user forgets to type NEW and press RETURN, his program may not work properly. Line numbers from a previous program may merge with the new program and produce unexpected results.

SAVE: stores a program on a disk. Type the word SAVE followed by the name of the program and press RETURN. The program will be saved to a disk. Its title may be seen in the catalog listing of disk contents. (Example: SAVE GAMES)

LOAD: brings a program from a disk into memory. Type the word LOAD followed by the name of the program and press RETURN. The program will be loaded into memory. The user then types RUN and presses RETURN to execute the program. (Example: LOAD GAMES)

CATALOG: displays the names of files stored on a disk. Type the word CATALOG and press RETURN.

DELETE: removes a file from a disk. Type the word DELETE followed by the name of the program and press RETURN. (Example: DELETE GAMES)

INFORMATION YOU SHOULD KNOW

1. There are different versions of BASIC, but all have common characteristics. Apple computers use Applesoft BASIC and Integer BASIC.

2. When you type a BASIC program, it goes into the computer's memory.

3. When you turn off the computer, the computer loses whatever was in the random access memory. (RAM)

4. Programs can be saved on cassette tapes or floppy disks. Then they do not have to be retyped.

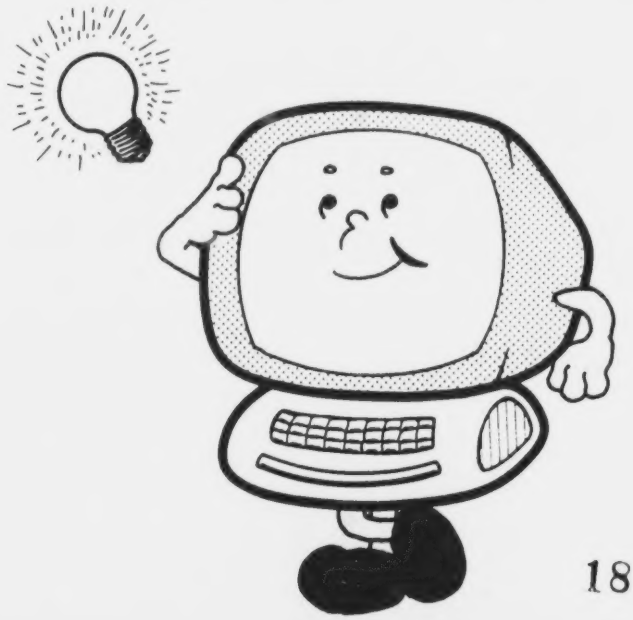
5. You should type the NEW command when you are ready to type a different program. NEW clears the first program out of memory. If you do not type NEW, you may get parts of both programs mixed together.

6. When you type RUN to execute a program, the computer goes to the lowest line number first. It follows the instructions and goes to the next higher number unless it is directed to a different line number.

7. If you need to see the entire listing of a program, type LIST and press the RETURN key. The list will be displayed.

8. If you need to stop a program, hold down the CONTROL key and press key C. The computer stops the program and tells you which line in the program was interrupted. After Control-C has been pressed, you must type RUN to get the program to execute again.

9. You may stop the listing of a program as it scrolls by on the screen by pressing CONTROL-S. To continue the scrolling of the listing, press any key.



18

SYNTAX ERROR

What did the crow say when he got a syntax error message? RUN this program and see.

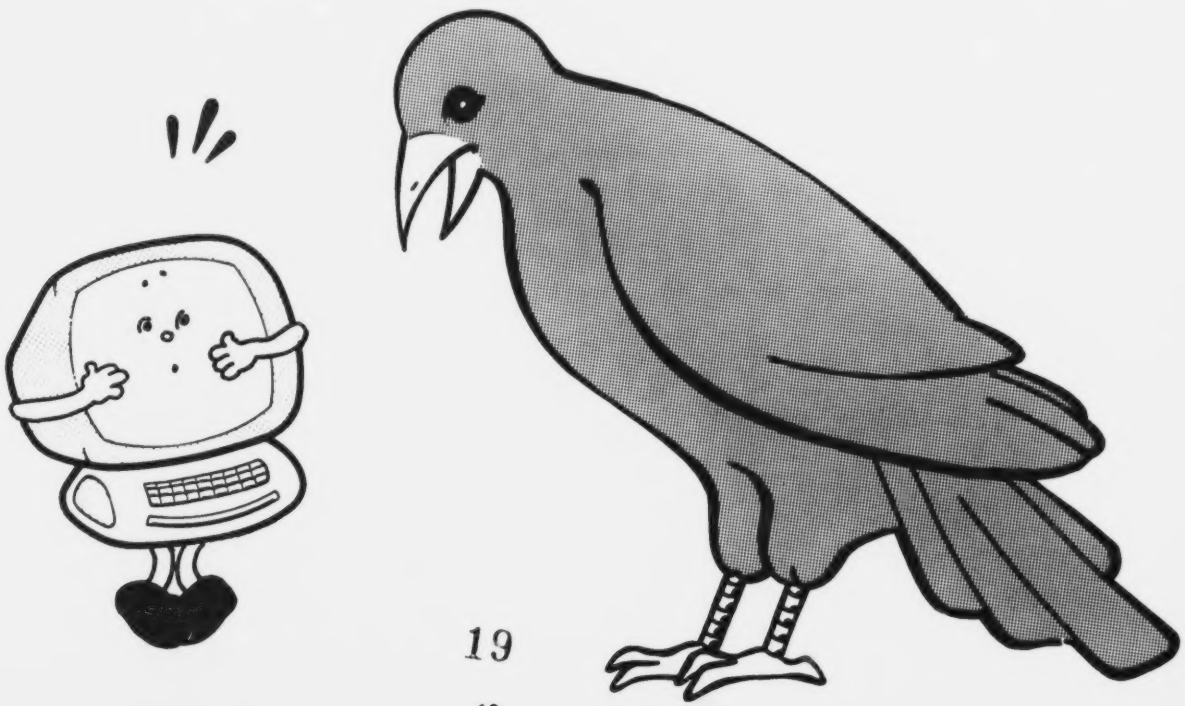
```
1Ø A$="FOR"
2Ø B$="CAWS"
3Ø C$="NO"
4Ø D$="ALARM"
5Ø PRINT C$" "B$" "A$" "D$"."
6Ø END   (space) (space) (space)
RUN
```

A syntax error means the computer did not understand what you typed. If the computer responds with "SYNTAX ERROR", the message means a mistake has been made. You must retype the line correctly.

A syntax error may show up for different reasons. The most common reasons are:

1. Incorrect spelling
2. Punctuation mistake
3. Omission of a keyword

What should you do about a syntax error? Just retype the line number that contains the error. Type it correctly and RUN to see if the correction was made.



CORRECTING MISTAKES

Sometimes a mistake will be made in typing a line. Suppose you realize the error before pressing RETURN. If your computer is an Apple IIe or IIc, press the left-arrow key and the cursor will move one space to the left. Press it as many times as necessary to back up to the position where the mistake occurred. Retype the line correctly and press RETURN. Your line will be corrected.

The right-arrow key on Apple IIe and IIc can be used to move over a character that you want to keep. Pressing the right-arrow key to move over previously typed characters is the same as retyping those same characters.

Older Apple computers do not have the arrow keys. Editing is done on them by pressing ESC and using the I, J, K, and M keys. I moves up, J moves left, K moves right, and M moves down. Press ESC once and then press I, J, K, or M to move to the appropriate space. Then press the space bar one time and you are ready to edit the line.

Try correcting this line. Type it exactly as you see it.

```
10 PRINT "I LOKE COM
```

Stop. Now back the cursor up to the O in LOKE. Type IKE. Then type a space and the word COMPUTERS. Add a quotation mark and press RETURN.

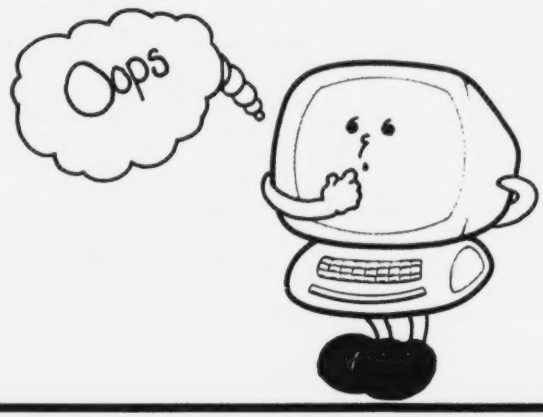
Now correct this mistake.

```
100 PRINT "MISAKES ARE COMMON."
```

Suppose you had pressed RETURN before you realized you had misspelled MiSTAKES. How would you correct the mistake?

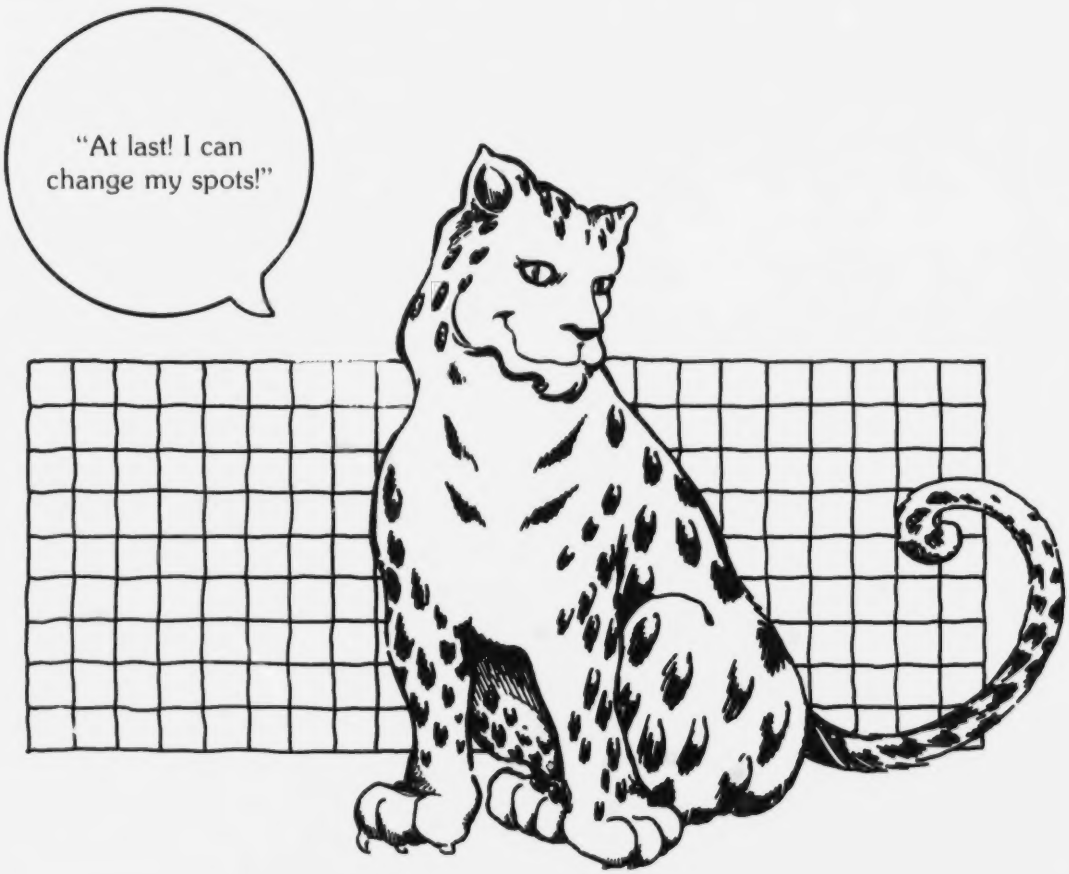
If you do not catch your error before pressing RETURN, some errors will be noted by the computer's SYNTAX ERROR reply when the program is RUN. Spelling mistakes inside quotation marks will not be called to your attention.

Mistakes can be corrected by retyping a line number and the information you want it to contain. Press RETURN and the mistake will be replaced with the newly typed line.



CHANGING A LINE

```
10 SPEED = 100
20 PRINT "SPOTS, SPOTS, SPOTS"
30 GOTO 20
RUN
```



A line can easily be changed after a program is written. Simply retype the line number and type the line the way it should be. Then press RETURN. Presto! You have your correction.

Type and RUN the program at the top of the page. Then retype line 20 to read "STRIPES." RUN the program. What happened?

NOTE: Type CONTROL-C to stop the program.

ON YOUR OWN: Experiment a little. Type a short program. Then change a line in the program. RUN it and see the result.

DELETING A LINE

Type this program. Do you understand how this program will work? RUN it and see if you are right.

```
10 ? "SMILE"
20 ? "A SMILE"
30 ? "IT'S CONTAGIOUS!"
```

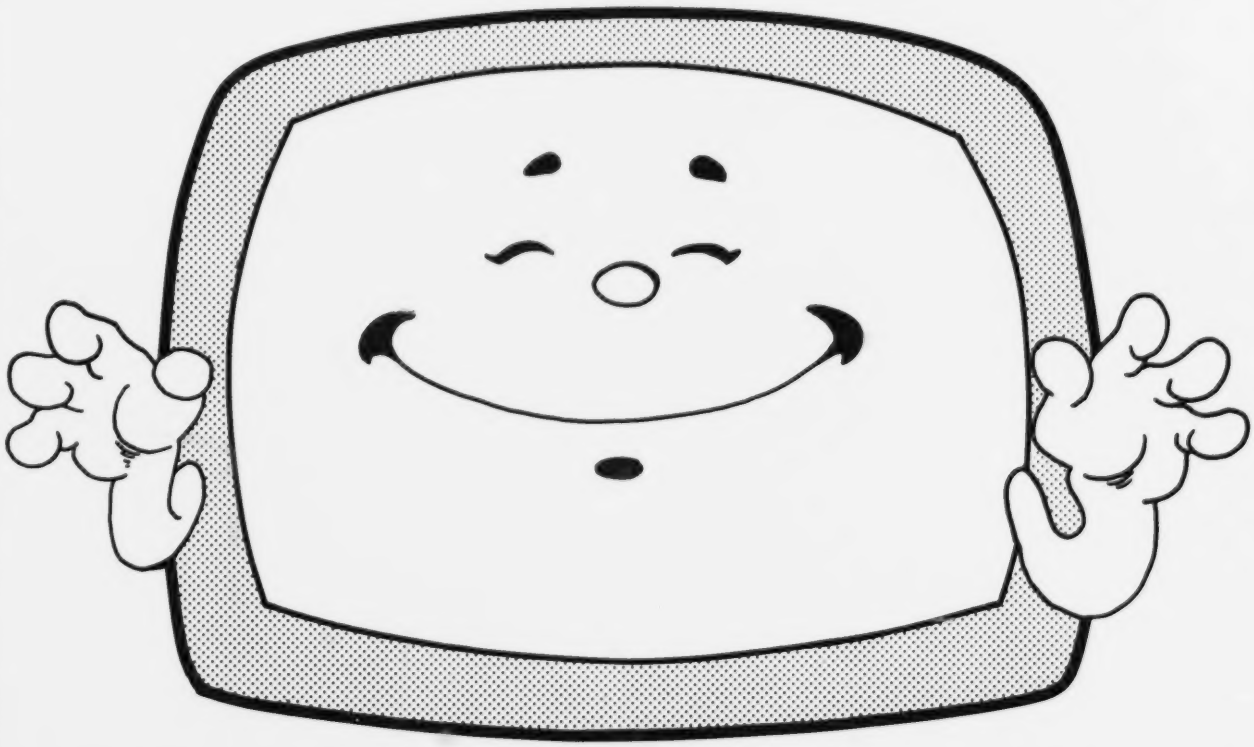
A line can be removed or deleted from a program very easily. All you do is type the line number and press the RETURN key. The entire line will disappear.

Type 20 and press RETURN.

Now RUN the program again. How is it different?

Do you remember how to see a program listing? Type LIST and press RETURN. The line numbers and their instructions will be displayed. Do that now and see if line 20 has been removed from your program.

Now delete line 30. What happens when you RUN the program after removing line 30?



A BASIC LOOP

Let's type and RUN another simple program in BASIC. First, type NEW and press RETURN. NEW is a command that clears a program out of the computer's memory so a different program can be entered. It is important to clear the memory before typing another program. If you do not clear the memory, parts of both programs may become merged. The results are often chaotic.

Here is a program similar to the first one you typed. Try this one and see what happens:

```
10 PRINT "WHIZ KID"
20 GOTO 10
30 END
```

Now type RUN and press RETURN. What happened?

The program can be stopped by pressing CONTROL-C.

The GOTO keyword told the computer to go back to line 10 and print WHIZ KID. It did and went on to line 20 where it again was directed back to the first statement. This program is an example of an infinite loop. Line 30 is never reached.

Can you guess what a loop does? It automatically repeats itself. That is why your program continued to print WHIZ KID. You escaped from the loop by using a break. (CONTROL-C)

Try another loop with this program:

```
10 PRINT "I'M REPEATING MYSELF."
20 GOTO 10
30 END
RUN
```

ON YOUR OWN: Write a program that contains a loop.

EXPERIMENTING WITH GOTO

Type and RUN these programs to see what GOTO tells the computer to do. Remember to use CONTROL-C to stop the program.

```
10 SPEED = 100
20 PRINT "I LIKE APPLES."
30 PRINT "YOU LIKE APPLES."
40 PRINT "WHO LIKES APPLES?"
50 GOTO 20
```

Line 10 slows the speed of the display so it can be read. Normal speed is 255, but that is too fast to be read easily. You may want to experiment with different speeds. You may change line 10 to SPEED = 255 to restore the normal speed.

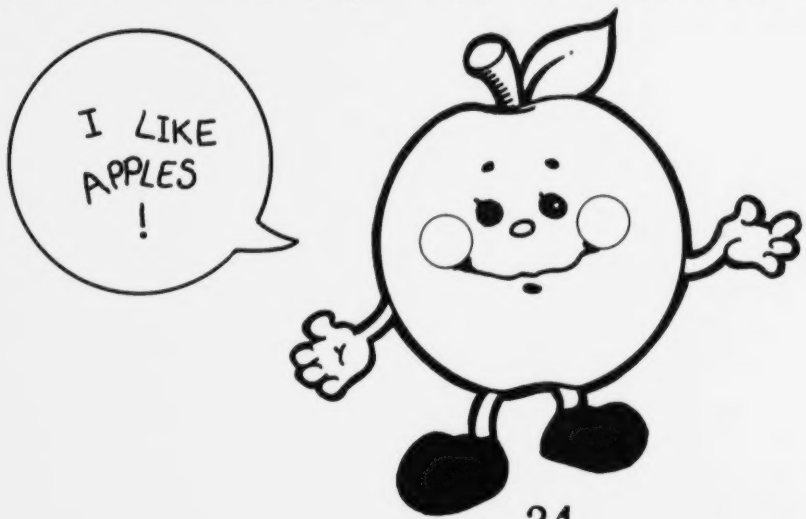
```
NEW
10 PRINT "I'M GOING IN"
20 PRINT "CIRCLES"
30 GOTO 20
```

```
NEW
10 PRINT "GOTO REPEATS"
20 PRINT "AND REPEATS"
30 GOTO 20
```

What do you think will happen to your program if a GOTO statement is used in the middle of the program?

What happens to the part of the program that follows the GOTO statement?

ON YOUR OWN: Write a short program of your own that utilizes a GOTO statement.



24

TO QUOTE OR NOT TO QUOTE

Quotation marks have a special purpose in BASIC statements. Here is an activity that will help you understand the purpose of quotation marks in a program.

1. Type PRINT 2 + 3 and press RETURN. What happened?

   _____

2. Type PRINT "2 + 3" and press RETURN. What happened?

   _____

Without quotation marks, the computer acts like a calculator. It prints the answers, in this case a 5. With quotation marks, the computer prints the exact message that is enclosed in quotation marks. In the example, 2+3 is printed.

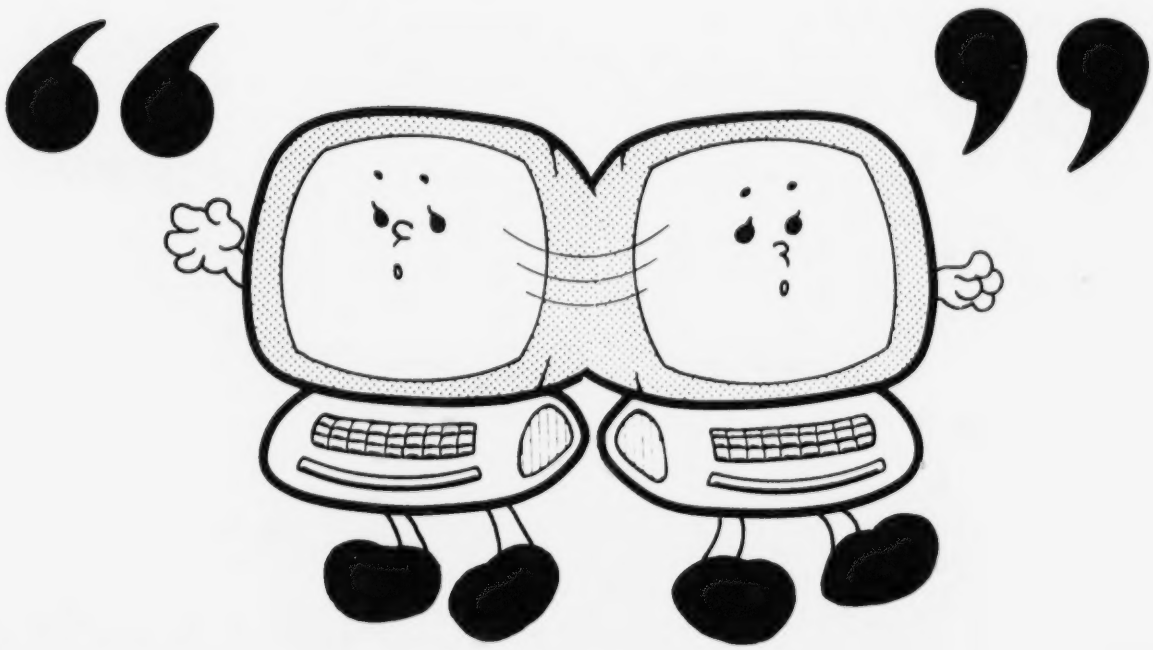
3. Type PRINT "BUSY PERSON" and RUN it. What happened?

   _____

4. Type BUSY PERSON and RUN it. Why do you get a syntax error message?

   _____

ON YOUR OWN: Write a simple program that does not use quotation marks. Remember to use a line number to begin each statement. Write a simple program that uses quotation marks.



MORE ABOUT QUOTATION MARKS

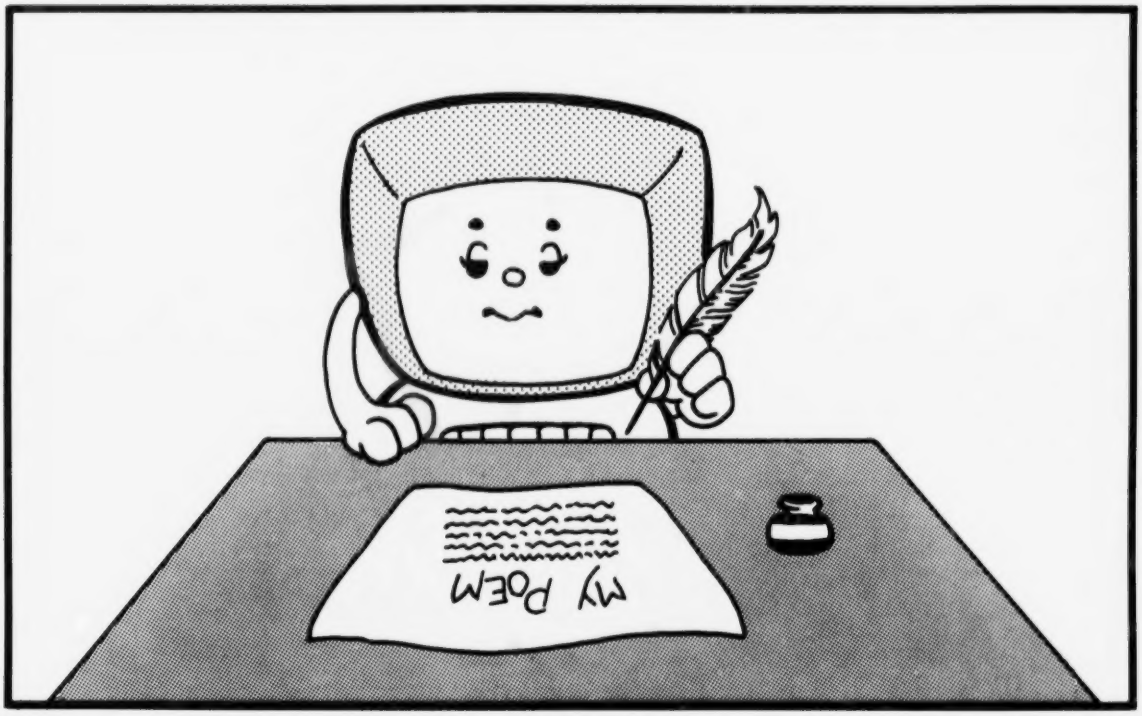
As you have learned, quotation marks are used in a program statement to tell the computer exactly what is to be printed.

Suppose you want to type a poem in BASIC. Each line of the poem should be assigned a different line number. Each line of the verse should be enclosed with quotation marks.

Type this poem exactly as you see it. It was written by Christina Rossetti and is entitled "Who Has Seen the Wind?"

```
 10 REM WHO HAS SEEN THE WIND?
 20 HOME
 30 PRINT "WHO HAS SEEN THE WIND?":PRINT
 40 PRINT "WHO HAS SEEN THE WIND?"
 50 PRINT "NEITHER I NOR YOU;"
 60 PRINT "BUT WHEN THE LEAVES HANG TREMBLING,"
 70 PRINT "THE WIND IS PASSING THROUGH."
 80 PRINT
 90 PRINT "WHO HAS SEEN THE WIND?"
100 PRINT "NEITHER YOU NOR I;"
110 PRINT "BUT WHEN THE TREES BOW DOWN THEIR HEADS,"
120 PRINT "THE WIND IS PASSING BY."
RUN
```

ON YOUR OWN: Are you ready to program a poem of your own? Type and RUN a short verse. If you write more than one verse, you may wish to include a line number with the word PRINT between each verse. That will leave a blank line between the verses.



BASIC INPUT

Here is a BASIC program that uses INPUT. An INPUT statement is used to get information from the person at the computer keyboard. Type the program below exactly as you see it. Then type RUN and press RETURN.

```
10 HOME
20 INPUT X
30 INPUT Y
40 INPUT Z
50 LET A=X+Y+Z
60 PRINT A
70 END
```

When the program is RUN, a question mark (?) appears as the input prompt for X. That is your cue to type a number value for X. Type 3. Press RETURN. You will see another ? for input Y. Type 4. Press RETURN. Another ? awaits your input for Z. Type 5 for Z. Press RETURN.

What does the computer display?_____

RUN the program again with different input numbers. What happens?

_____

Can you type more than one number for each input? What do you think will happen? For

X, type 8, 5, 3. What does the computer do? _____

Try this: Type NEW. Then type the four lines that follow.

```
10 INPUT X,Y,Z
20 A=X+Y+Z
30 PRINT A
40 END
RUN
```

What happens if you type three numbers when the input prompt appears?

_____

What happens if you type five numbers?

_____

PROGRAMS WITH INPUT

Here is a short program that asks your name. Type and RUN it.

```
10 HOME
20 PRINT "WHAT IS YOUR NAME?"
30 INPUT N$
40 PRINT "HELLO, "N$;"."
50 END
```

(space)

Can you write a program that requests your address and telephone number?

Here is a program to forecast your lucky number.

```
10 REM YOUR LUCKY NUMBER
20 HOME
30 PRINT "THIS PROGRAM WILL TELL YOU YOUR LUCKY NUMBER."
40 PRINT "WHAT IS YOUR NAME?";N$
50 INPUT N$
60 PRINT
70 PRINT "TYPE ONE OR TWO DIGITS TO TELL ME THE MONTH OF YOUR
   BIRTH."
80 INPUT X
90 PRINT
100 PRINT "TYPE ONE OR TWO DIGITS TO TELL ME THE DATE OF YOUR
   BIRTH."
110 INPUT Y
120 PRINT
130 PRINT "TYPE TWO DIGITS TO TELL ME THE YEAR OF YOUR BIRTH."
140 INPUT Z
150 A = X + Y + Z
160 PRINT
170 PRINT N$;" YOUR LUCKY NUMBER IS ";A;"."
180 END
```

(space) (space)

VARIABLES

There are special storage places in the memory of your computer. You can save a number or a word in a storage space. A variable represents the place in the computer's memory where a certain value is stored.

Suppose you want to save the number 12. You can assign the number 12 to the letter N or any other letter. Then type PRINT N and press RETURN. Try it and see what happens.

```
1Ø N = 12
2Ø PRINT N
RUN
```

A variable must start with a letter. These are examples of variables:

C = 23

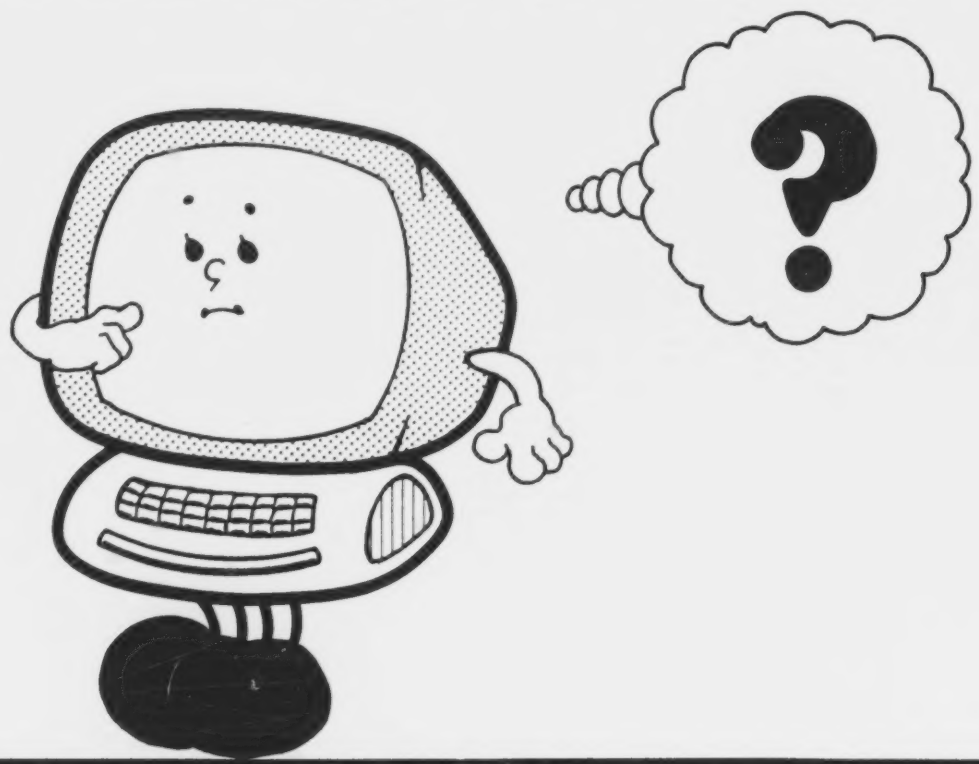
D = 4Ø

HOT = 1Ø

SUPER = 25

What happens if you assign variable C a value of 15 and then type C = 35? Try and see. Record your answer.

ON YOUR OWN: Write some variables of your own. Try them out with your computer.



MORE ABOUT VARIABLES

A variable is a symbol that represents a place in the computer's memory. It is not necessary to write a program with line numbers to demonstrate the use of variables. Suppose you type a word such as HAPPY and assign it a value of 3.

Type HAPPY = 3
PRINT HAPPY.

Now press RETURN. What does the computer do? _____

Type PRINT HAPPY + 6. Press RETURN.

What does the computer do? _____

Study the lines below. What do you think the computer will display when the program is RUN? Try and see.

```
HAPPY = 3
PRINT HAPPY
HAPPY = 5 + HAPPY
PRINT HAPPY
HAPPY = HAPPY - 2
PRINT HAPPY
HAPPY = HAPPY + HAPPY
PRINT HAPPY
```

How did the computer respond as you typed each line?

_____

What does this demonstrate?

_____

Compare the results with the same information written in program lines.

```
1Ø HAPPY = 3
2Ø PRINT HAPPY
3Ø HAPPY = 5 + HAPPY
4Ø PRINT HAPPY
5Ø HAPPY = HAPPY - 2
6Ø PRINT HAPPY
7Ø HAPPY = HAPPY + HAPPY
8Ø PRINT HAPPY
RUN
```

ON YOUR OWN: Experiment with variables. Assign some values of your own and manipulate them in different ways.

LET

LET is a keyword that gives a specific value to a numeric variable or to a string variable. A numeric variable contains a number value, such as X = 4. A string variable contains text characters (letters, punctuation marks, or digits), and it must end with a $. Example: B$ = 5

Study this sample program.

```
10 LET D=4
20 LET R=5
30 LET S=D * R
40 PRINT S
50 END
```

What do you think will happen when you RUN the program? RUN it and see.

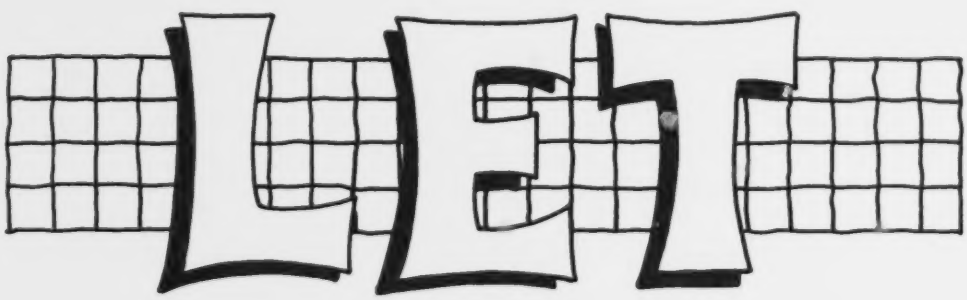
Study this program. What do you think it will do?

```
10 LET J=20
20 LET R=30
30 LET F=R - J
40 PRINT F
50 END
```

Type the following program. As you type, be sure to leave a space before the quotation mark after the words YOU, ARE, and A. That is so the spacing will be correct between the words when the message is printed.

```
                    (space)
10 LET A$= "YOU↓"
20 LET B$= "WHIZ!"
30 LET C$= "ARE↖"
40 LET D$="A ↙"  (space)
50 PRINT A$ + C$ + D$ + B$
60 END
```

Write what the computer displays when the program is RUN.



STRING FEVER

A string is any character or set of characters inside quotation marks. Because the characters are inside quotation marks, the computer prints them exactly as typed. The quotation marks are not part of the string.

$ means string. (The symbol is not a dollar sign even though that is the symbol that is typed from the keyboard.)

Try this.

```
10 LET A$="FANTASTIC"
20 LET B$="JUST"
30 PRINT A$;B$;A$
40 END
```

RUN the program and record the results. _____

_____

Did the words run together? Now type the program this way. Allow a space after each word in quotation marks before putting the closing quotation mark.

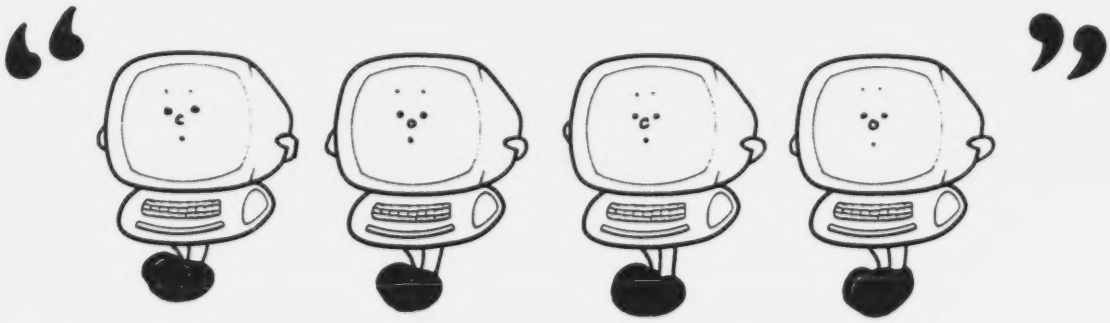
```
10 LET A$="FANTASTIC "
20 LET B$="JUST "
30 PRINT A$;B$;A$
40 END
```

(space)

How is the spacing this time?_____

Remember, a string variable holds whatever letters or characters are typed inside quotation marks. The string variable must end with $. When the string variable is needed, its letter and $ are typed. Any character recognized by BASIC may be in a string except the quotation mark("). The string variable calls forth the characters that are stored in it.

A string may hold from 0 to 255 characters. If there are no characters between two quotation marks, the string is said to be null.



REM

REM means remark. A remark statement is not printed when the program is executed. Its purpose is to give information to someone reading a program listing or writing a program. It explains what a line or a section of a program does.

EXAMPLE: 10 REM MATH SYMBOLS

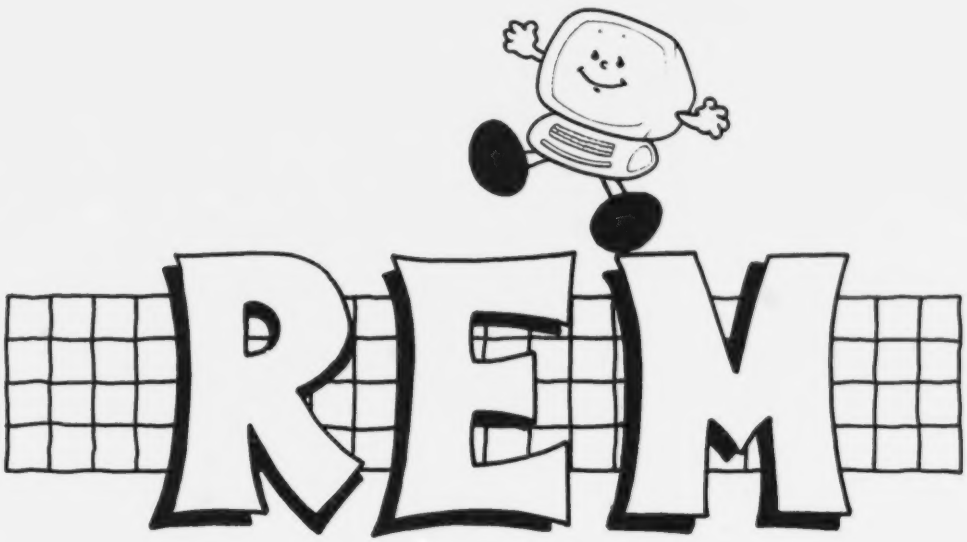
This remark tells the reader that the next section of the program concerns math symbols. There is nothing for the computer to perform in this line. It simply makes it easier for the reader to know what the program contains.

Here is a program with three remarks. Type the program and RUN it. Then type LIST to see that the REM statements are a part of the program listing.

```
 10 REM AVERAGING SIX SPELLING GRADES
 20 HOME
 30 PRINT "THIS PROGRAM WILL FIND THE"
 40 PRINT "AVERAGE OF SIX SPELLING GRADES."
 50 PRINT "EACH TIME YOU SEE A QUESTION MARK,"
 60 PRINT "TYPE A GRADE AND PRESS RETURN."
 70 PRINT
 80 INPUT A,B,C,D,E,F
 90 REM COMPUTE THE AVERAGE
100 LET Z=(A+B+C+D+E+F)/6
110 REM PRINT THE AVERAGE
120 PRINT
130 PRINT "THE AVERAGE IS ";Z;"."
140 END
```

(space)

ON YOUR OWN: Write a program that includes at least two remarks.



USING CONCATENATION

Concatenate (kän-kat′-e-nāt) means to link together. You can use a plus sign (+) to instruct the computer to put strings together. That is called concatenation.

Try this example:

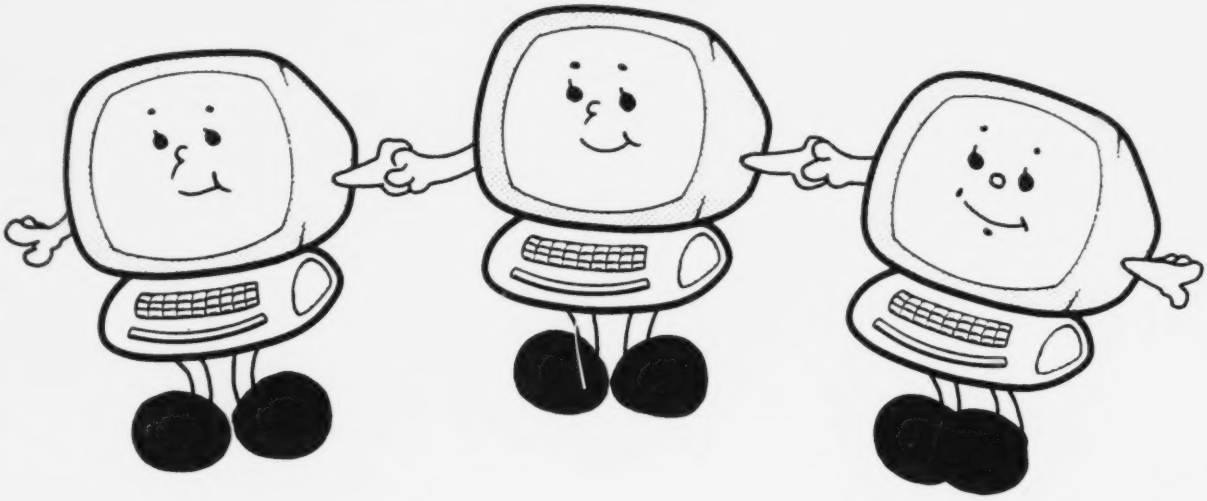
```
10 A$ = "CUP"
20 B$ = "CAKE"
30 PRINT A$ + B$
40 END
RUN
```

What is the output? _____

You can store concatenated strings in address locations. For example, C$ holds the concatenated string of A$ and B$.

```
10 A$ = "CUP"
20 B$ = "CAKE"
30 C$ = A$ + B$
40 PRINT C$
50 END
```

ON YOUR OWN: Write a short program that will concatenate your first, middle, and last names. Remember to allow a space after your first name and after your middle name so your complete name will be spaced correctly when the three strings are concatenated.



PROGRAMS WITH CONCATENATION

Here are two programs that show how strings can be concatenated.

```
 10 REM SILLY SENTENCES
 20 HOME
 30 LET A$ = "THE FAT FROG" (space)
 40 LET B$ = "ATE"
 50 LET C$ = "THE TINY FLEA"
 60 LET D$ = A$ + B$ + C$
 70 LET E$ = C$ + B$ + A$
 80 PRINT D$;"."
 90 PRINT
100 FOR I=1 to 3000:NEXT I
110 PRINT "NO, ";E$;"!"
          (space)
NEW
 10 DIM A$(25), B$(25), C$(25), D$(25)
 20 PRINT
 30 PRINT "TYPE A NAME."
 40 INPUT A$
 50 PRINT "TYPE AN ADJECTIVE."
 60 INPUT B$
 70 PRINT "TYPE A NOUN."
 80 INPUT C$
 90 PRINT "TYPE A NUMBER LARGER THAN 1."
100 INPUT D$
110 PRINT
120 PRINT "HAVE YOU HEARD THE LATEST?"
130 PRINT
140 PRINT A$;" JUST WON A BIG ";B$;" ";C$;" WORTH ";D$;
    " DOLLARS." (space)          (space) (space) (space) (space)
  (space)
```

NOTE: If the concatenated string is longer than 255 characters, the program will stop. A ? STRING TOO LONG ERROR will be displayed.

IF THEN

One important capability of BASIC is its ability to compare and decide. Programmers call this feature "conditional branching."

Remember the GOTO? It directed the computer to a different line number in a program. That is an example of branching.

Conditional branching does the same thing, except it only goes to a specific line number under certain conditions. The computer has to compare and decide. In BASIC the IF THEN statement instructs the computer to compare and decide.

Study the explanation below.

```
200 IF (a condition) THEN (an action)
```

How does the computer compare and decide? The computer reads a statement and checks to see if it is true or false. If it is true, the computer branches to a specified line number. If it is false, the computer continues on to the next line number in the program.

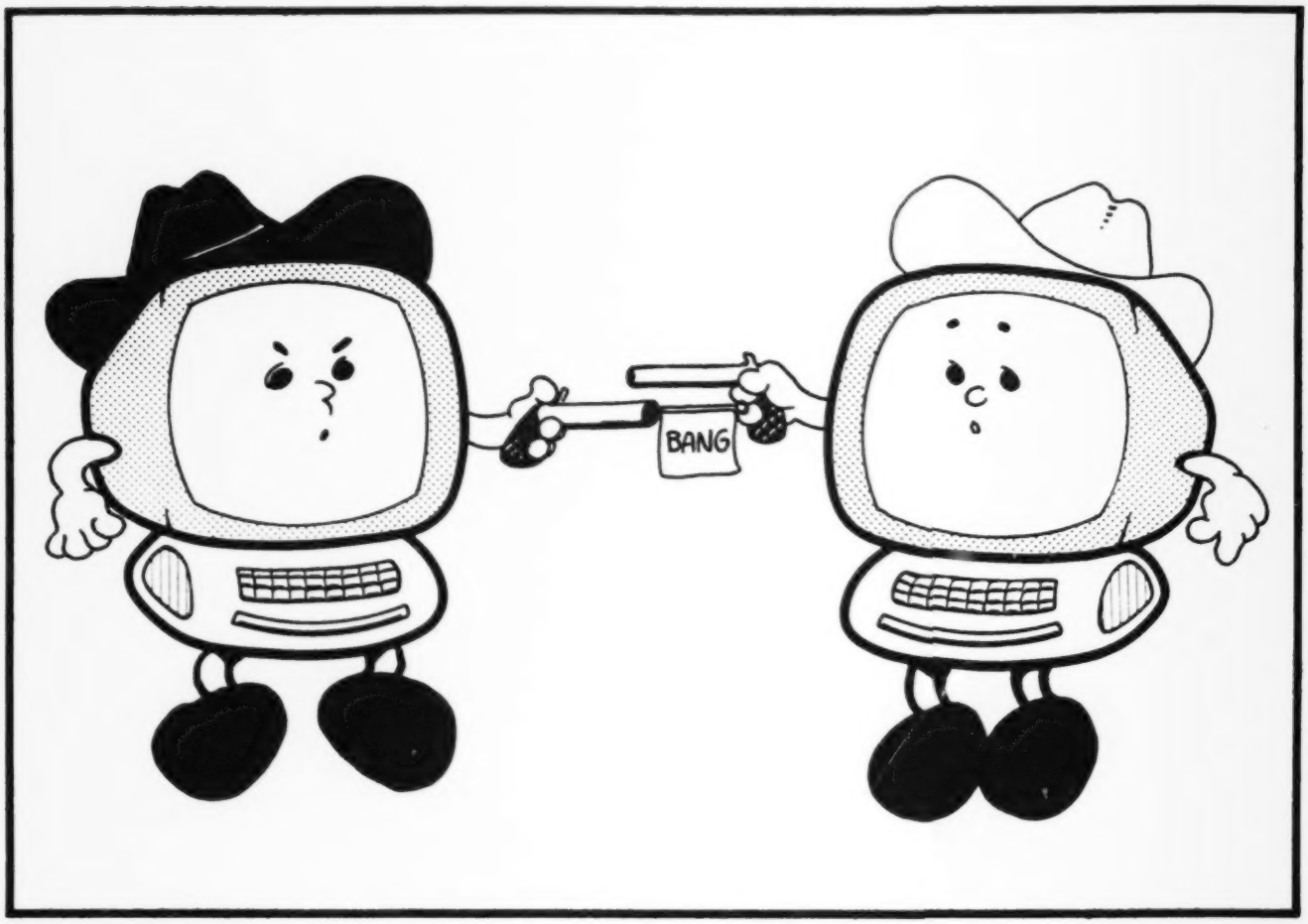
EXAMPLE:

```
 10 HOME
 20 PRINT "TYPE A DIGIT FROM 1 TO 5."
 30 INPUT D
 40 IF D=1 THEN PRINT "APPLES"
 50 IF D=2 THEN PRINT "ORANGES"
 60 IF D=3 THEN PRINT "PEARS"
 70 IF D=4 THEN PRINT "GRAPES"
 80 IF D=5 THEN PRINT "BANANAS"
 90 PRINT
100 GOTO 20
110 END
RUN
```

AN IF . . . THEN PROGRAM

```
 10 HOME
 20 PRINT "PICK A NUMBER, 1 OR 2. TYPE YOUR CHOICE."
 30 INPUT X
 40 IF X = 1 THEN 100
 50 IF X = 2 THEN 200
 60 IF X <> 1 AND X <> 2 THEN 20
100 PRINT "YOU'RE GUN TO LIKE PROGRAMMING."
110 PRINT "IT WILL GET YOU ALL FIRED UP!"
120 PRINT
130 PRINT "YOU HOT SHOT!"
150 END
200 PRINT "YOUR PROGRAMMING IS RIGHT ON TARGET."
210 PRINT
220 PRINT "YOU COOL HIT."
230 END
RUN
```

ON YOUR OWN: Write a program that uses conditional branching. Share your program with a friend.

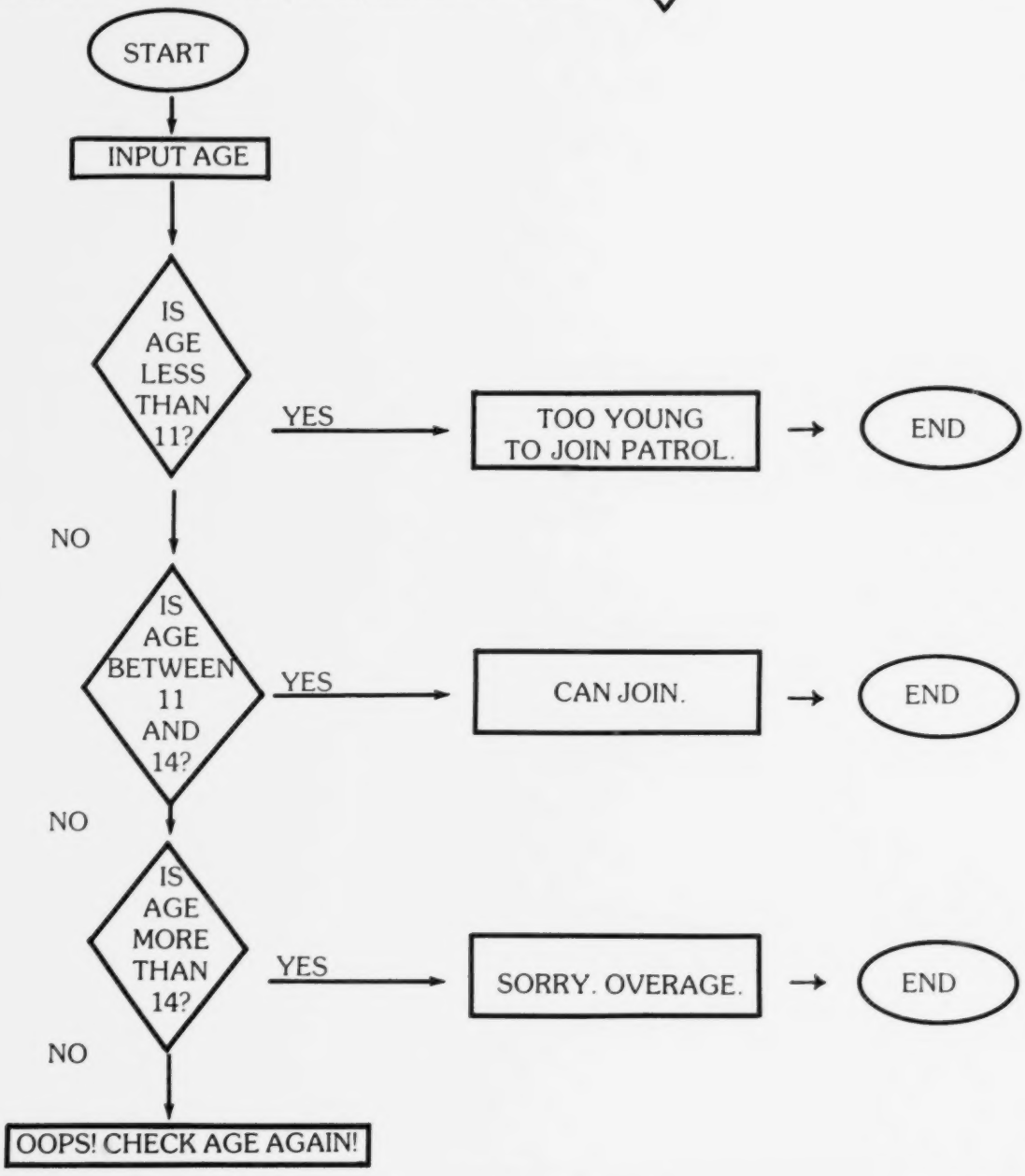


READING A FLOWCHART

To join the school safety patrol a student must be between the ages of 11 and 14. The flowchart illustrates how the computer reaches a decision concerning whether or not an individual can join the patrol.

This is a decision box. A yes or no decision must be made when you see it.

Study the flowchart. Can you explain how it works?

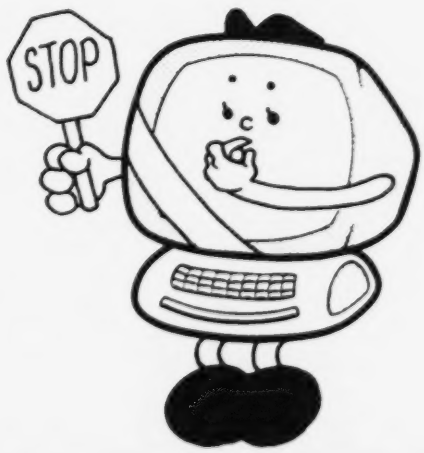


AN IF THEN PROGRAM: SAFETY PATROL

Here is a program based on the flowchart to determine who may join the safety patrol. Type and RUN the program. Does it work the same as the flowchart?

```
 10 REM SCHOOL SAFETY PATROL
 20 HOME
 30 PRINT "TO JOIN THE SCHOOL SAFETY PATROL"
 40 PRINT "A STUDENT MUST BE"
 50 PRINT "AT LEAST ELEVEN YEARS OLD"
 60 PRINT "AND NOT MORE THAN FOURTEEN YEARS OLD."
 70 PRINT
 80 PRINT "HOW OLD IS THE STUDENT?"
 90 INPUT A
100 IF A = 11 THEN 160
110 IF A = 12 THEN 160
120 IF A = 13 THEN 160
130 IF A = 14 THEN 160
140 IF A < 11 THEN 180
150 IF A > 14 THEN 200
160 PRINT "STUDENT MAY JOIN SAFETY PATROL."
170 GOTO 210
180 PRINT "CAN'T JOIN YET. TOO YOUNG."
190 GOTO 210
200 PRINT "SORRY. OVERAGE."
210 PRINT
220 PRINT "HOW OLD IS THE NEXT STUDENT?"
230 GOTO 90
240 END
```

NOTE: Type CONTROL-C to exit the program.



FOR NEXT

Study this program.

```
10 HOME
20 FOR N = 1 TO 10
30 PRINT "N =↑"; N
40 NEXT N  (space)
50 PRINT "THAT'S ALL!"
```

Here is how the program works. Line 10 moves the cursor to the top left corner of the screen. Line 20 assigns N a value of 1. In Line 30 the 1 is printed. Line 40 sends the computer back to Line 20 where the value of N is increased by 1. Line 30 prints 2, and Line 40 continues the loop back to Line 20. After the loop has been carried out ten times, the computer goes to Line 50 and prints, "THAT'S ALL!"

Here is another program that demonstrates the FOR NEXT statement. Study the program and answer the three questions below.

```
NEW
100 FOR W = 1 to 100
110 PRINT W
120 NEXT W
130 PRINT "I'M THROUGH COUNTING."
```

1. What line number sets the range of the variable W?

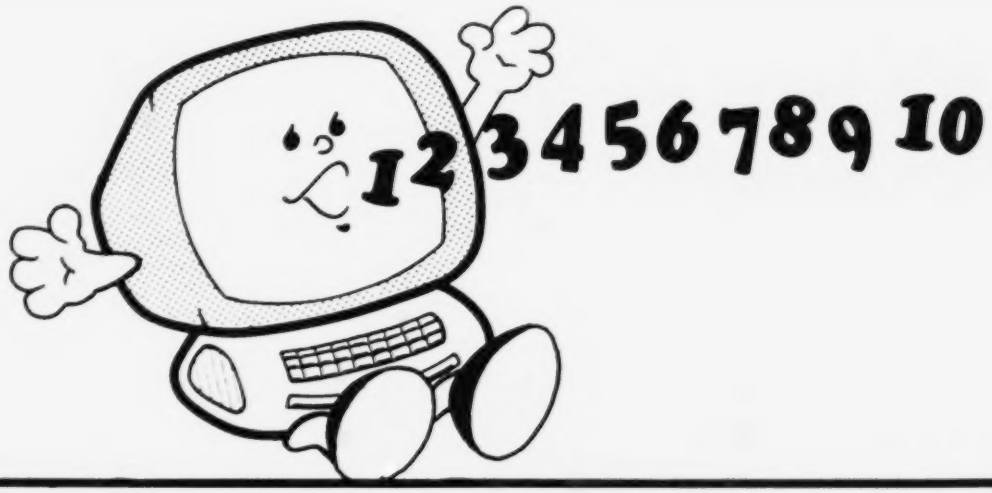
   _____

2. What line tells the computer to display W?

   _____

3. What happens when the range limit is reached?

   _____



READ AND DATA

READ and DATA are keywords that work together. READ looks for a DATA statement. Information is stored in a DATA statement. The stored information may be numbers or strings.

Study the program below. Be sure to type a space before the word COMPUTERS and after the word I in line 300. Type and RUN the program.

```
100 READ X$, Y$, Z$
200 PRINT Y$ + Z$ + X$
300 DATA " COMPUTERS.","I ","LIKE"
400 END
RUN
```

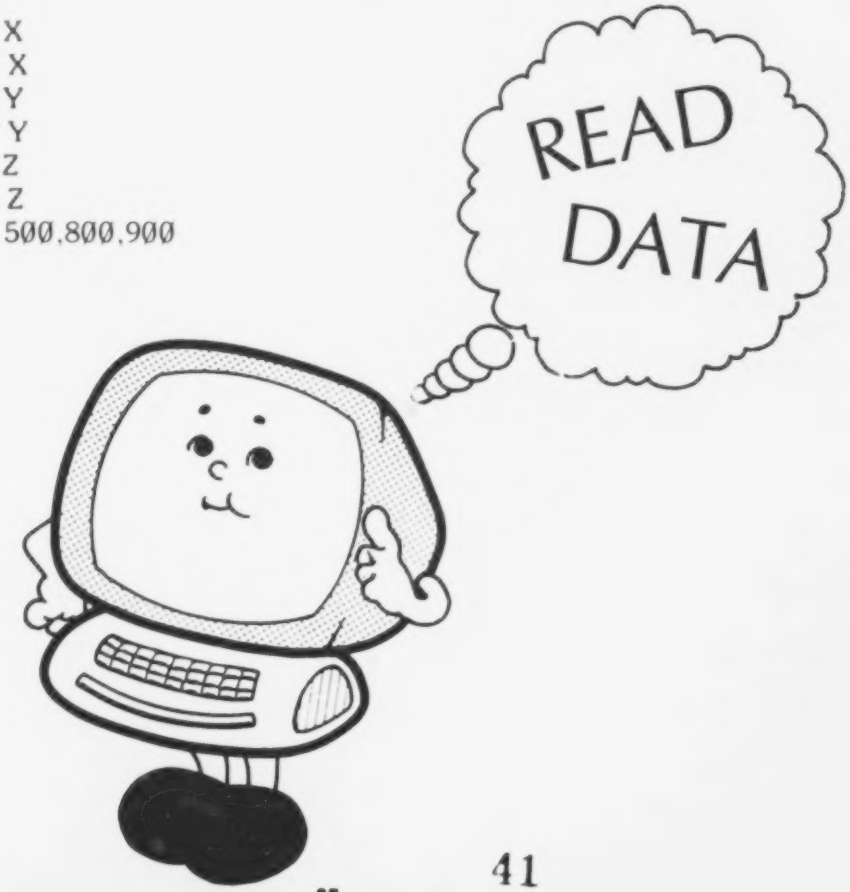
(space) (space)

Did you figure out what happened? The computer took the first item in the DATA statement and assigned it to the first variable in the READ statement. Thus, the word COMPUTERS was assigned to X$. The word I was assigned to Y$, and the word LIKE was assigned to Z$.

NOTE: If you want commas in a DATA string, you must place them inside quotation marks.

Type and RUN this program to see how READ DATA works:

```
NEW
10 READ X
20 PRINT X
30 READ Y
40 PRINT Y
50 READ Z
60 PRINT Z
70 DATA 500,800,900
80 END
RUN
```



MORE ABOUT READ AND DATA

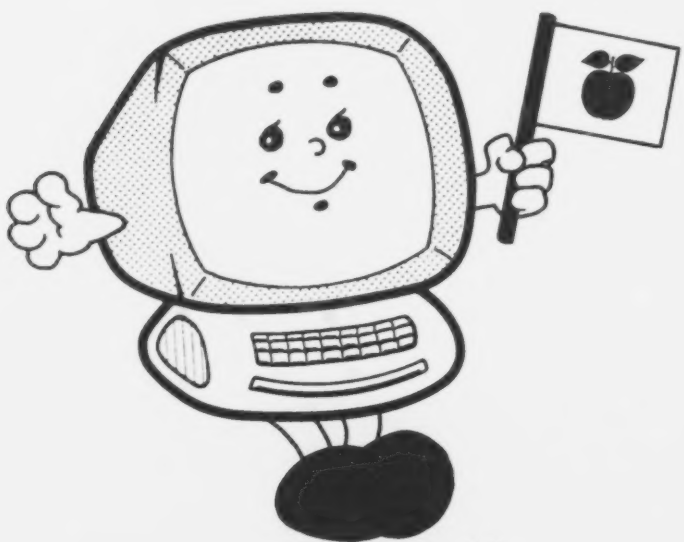
READ and DATA statements are used together to get information from within a program. A DATA statement contains a list of numbers or words separated by commas following the word DATA. A READ statement reads one or more items from the DATA list.

The DATA list is read in sequential order. That means the items are read in the order of their entry. The program cannot back up or skip over an item. However, the list of items can be read again with a RESTORE statement.

An OUT OF DATA ERROR will be displayed if the READ statement tries to read beyond the last item in the DATA statement. Line 3Ø in each program below prevents an OUT OF DATA ERROR by including a "flag" to end the reading. RUN each program to see how the flag works.

```
1Ø HOME
2Ø READ X
3Ø IF X=Ø THEN END
4Ø PRINT X
5Ø GOTO 2Ø
6Ø DATA 5, 1Ø, 15, 2Ø, 25, Ø
RUN
```

```
NEW
1Ø HOME
2Ø READ X$
3Ø IF X$="DONE" THEN END
4Ø PRINT X$
5Ø GOTO 2Ø
6Ø DATA "PENCILS, PAPER, ERASERS, INK"
7Ø DATA "NOTEBOOK, CRAYONS", "DONE"
RUN
```



RANDOM

RND means random. It is a function used to select numbers as if they were drawn out of a hat. There are different ways to use RND, but we will use it with the integer function(INT). The integer function gives us whole numbers.

Type and RUN the program below.

```
10 X = INT(8 * RND(1) )
20 PRINT X
30 FOR I = 1 TO 500:NEXT I
40 GOTO 10
```

Line 10 selects a random whole number from a choice of eight numerals, 0 to 7. Line 20 prints the number that was selected. Line 30 acts as a timer. The screen display remains unchanged for a count of 500, and line 40 forms a loop back to line 10 where another number is selected. The process continues until the user interrupts it by typing CONTROL-C.

You may wish to experiment with different timing counts in line 30. The number 500 may be changed to a larger number to hold the display on the screen longer or a smaller number to hold the display a shorter length of time.

Can you explain what happens in the program below?

```
10 X = INT(20 * RND(1) )
20 PRINT X
30 FOR I = 1 TO 500:NEXT I
40 GOTO 10
RUN
```

The following program produces random numbers from 0 to 9, but it differs from the programs above. It does not contain a loop or timing feature. It prints ten random numbers and stops.

```
10 FOR N = 1 to 10
20 PRINT INT(RND(1) * N)
30 NEXT N
40 END
```

Type and RUN the above program several more times to see how the computer selects and prints different numbers from 0 to 9.

ON YOUR OWN: Write a program that prints random whole numbers from 0 to 25. Remember to use the INT function with RND(1).

A RND PROGRAM: GUESS MY NUMBER

```
10 REM GUESS MY NUMBER
20 PRINT "CAN YOU GUESS MY NUMBER?"
30 PRINT "THE NUMBER IS A WHOLE NUMBER FROM 1 to 25."
40 X = INT(24 * RND(1) + 1)
50 PRINT: PRINT "I'M READY. YOU MAY START GUESSING."
60 PRINT: INPUT "WHAT IS YOUR GUESS?";G
70 IF G = X THEN PRINT "TERRIFIC GUESS! LET'S PLAY THAT AGAIN.":
   FOR I = 1 TO 2000: NEXT I: HOME: GOTO 40
80 IF G◄X THEN PRINT "THINK BIGGER.":GOTO 60
90 IF G►X THEN PRINT "THINK SMALLER.":GOTO 60
100 END
```

QUESTIONS:

1. In line 50 what is the purpose of typing PRINT two times? _____

   _____

2. What does RND do? _____

   _____

3. Why doesn't line 10 print when the program is RUN? _____

   _____

4. Why doesn't the program reach line 100 when RUN? _____

   _____

LEN

LEN is a function in BASIC that causes the computer to print the number of characters in a string. A character is a letter, number, or symbol that can be typed from the keyboard. LEN is short for the word length.

Type and RUN this program to see how LEN works.

```
10 INPUT "YOUR LAST NAME";N$
20 PRINT N$;" HAS ";LEN(N$);" CHARACTERS."
```

(space) (space) (space)

Record what happened. _____

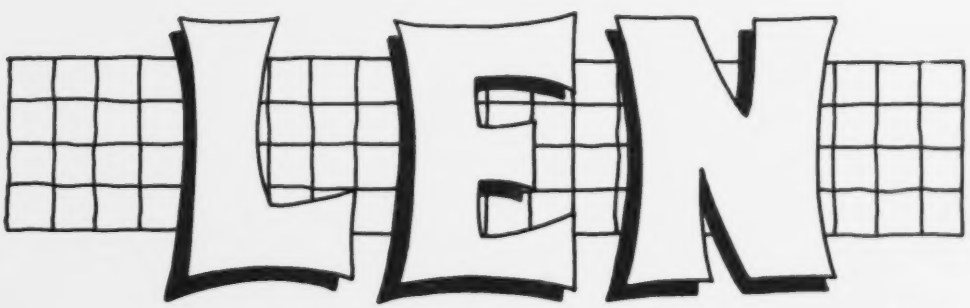
Remember, punctuation marks and spaces are included as characters.

Experiment with LEN. Try these entries:

```
10 PRINT LEN ("TURTLE")
RUN

10 PRINT LEN ("HOUSE")
RUN
```

You can also use a variable such as the letter N in this program to determine LEN.

```
10 N = LEN ("TURTLE")
20 PRINT N
RUN

10 PRINT LEN ("UNITED STATES")
RUN

10 PRINT LEN ("ALPHABET")
RUN
```



A LEN PROGRAM: ZAPALAND

```
10 REM ZAPALAND
20 HOME
30 PRINT "WHAT IS YOUR NAME?";N$
40 INPUT N$
50 PRINT
60 PRINT N$;",↑WELCOME TO ZAPALAND, THE ANIMAL PARADISE."
70 PRINT    (space)
80 PRINT "THE ANIMALS LOVE IT HERE."
90 PRINT
100 PRINT "YOU MAY FIND YOUR FAVORITE ANIMALS IN DIFFERENT
    PARTS OF ZAPALAND."
110 PRINT
120 PRINT N$;",↑WHAT ANIMAL WOULD YOU LIKE TO LOCATE?";A$
130 PRINT    (space)
140 INPUT A$
150 PRINT
160 IF LEN(A$) = 2 THEN 230
170 IF LEN(A$) = 3 THEN 240
180 IF LEN(A$) = 4 THEN 250
190 IF LEN(A$) = 5 THEN 260
200 IF LEN(A$) = 6 THEN 270
210 IF LEN(A$) = 7 THEN 280
220 IF LEN(A$) ◄ 2 OR LEN(A$) ► 7 THEN 290
230 PRINT "THAT ANIMAL IS IN SNEELA.":GOTO 300
240 PRINT "YOU MAY FIND THAT ANIMAL TAKING A NAP IN BAPA.":GOTO
    300
250 PRINT "THAT ANIMAL JUST LEFT LEZO FOR PURZEL.":GOTO 300
          (space)
260 PRINT N$;",↓THAT ANIMAL IS IN BIMPA.":GOTO 300
270 PRINT "THAT ANIMAL WAS HERE, BUT IT JUST LEFT.":GOTO 300
280 PRINT "THAT ANIMAL IS IN JIMPER.":GOTO 300
290 PRINT "YOU MAY FIND THAT ANIMAL IN FLEECEVILLE.":GOTO 300
300 END
```

Feel free to experiment with this program. Add LEN statements and change locations to help get an understanding of how the LEN function works.

SPACED OUT

Information that is printed on the screen when a program is RUN is called output. The output can be spaced closer together or further apart. You determine how it will be spaced. If you type four numbers separated by commas in a PRINT statement, the first number will be displayed in column 1. The second number will be displayed in column 17. The third number will be displayed in column 33. The fourth number will appear in column 1 on the next line of the screen display.

Type this program:

```
10 PRINT "3","5","7","4","6","2"
20 END
RUN
```

What does the output look like? _____

_____

Try these two programs:

```
10 PRINT "3";"5";"7";"4";"6";"2"
20 END
RUN

10 PRINT "3","5";"7","4";"6","2"
20 END
RUN
```

After executing these programs, you should be able to answer the questions below.

1. What does a semicolon tell the computer to do? _____

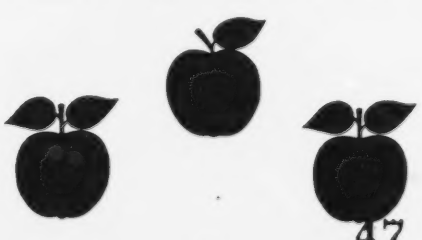
   _____

2. What does a comma tell the computer to do? _____

   _____

ON YOUR OWN: Write a program that places the names of six animals in different columns (tab positions). Do not use words that contain more than seven letters. The monitor screen is only 40 characters wide. (See page 43 for spacing words in other tab positions.)

Write the same program with the words separated by a single space.



TAB

TAB is used to print statements at specified locations. TAB ( ) tells the computer at what position to print something. A number goes inside the parentheses.

EXAMPLE:

PRINT TAB(15);"SSS"

This tells the computer to move the cursor to the 15th position in the line and print SSS.

Try this:

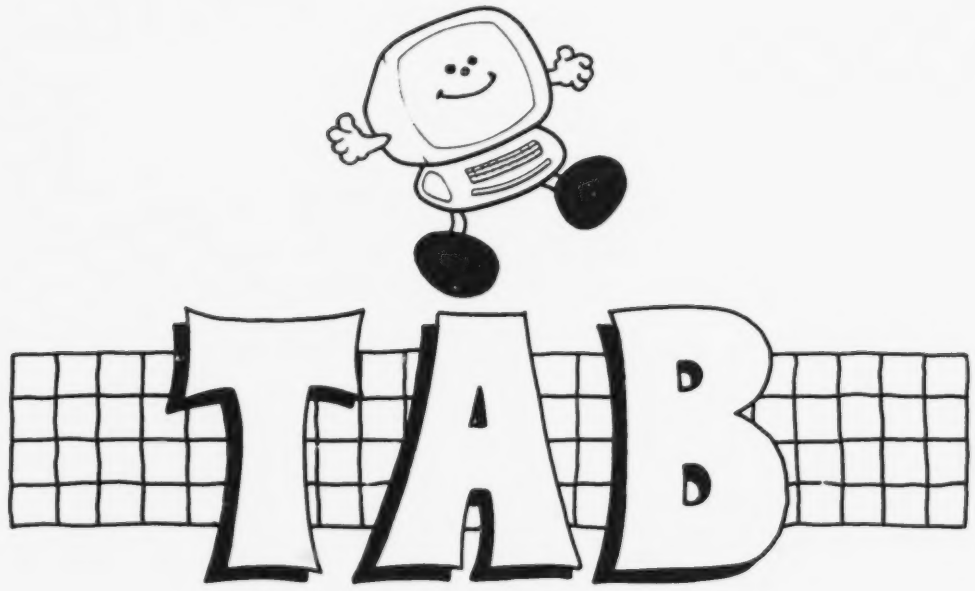
PRINT TAB(2Ø);"USA"

What happened? _____

TAB cannot space backwards. Therefore, you must carefully plan exactly where you want your printing to be displayed.

Now use TAB to space your initials in positions 5,1Ø, and 15.

ON YOUR OWN: Make a two-column display listing ten states and their capitals. Set the TAB in two positions of your choice.



HTAB AND VTAB

HTAB means horizontal tab. It is a statement used to move the cursor to a specific place in a horizontal direction. Unlike TAB, HTAB can move the cursor forward or backward.

HTAB 12 directs the computer to move 12 positions from the beginning of the horizontal line.

EXAMPLE:

```
 5 HOME
1Ø HTAB 12:PRINT "HELLO."
RUN
```

You may use HTAB to show more than one item on the same line, but you must type a semicolon at the end of each PRINT statement to keep the computer from typing a different line.

EXAMPLE:

```
 5 HOME                (space)
1Ø HTAB 14: PRINT "↓FUN!";
2Ø HTAB 5: PRINT "BASIC ";
3Ø HTAB 11: PRINT "IS"
```

If you will include a line 8 to slow the speed, you can see the order in which the words are displayed on the screen. Type 8 SPEED = 1Ø and RUN the program. To return the program to normal speed, change the speed in line 8 to 255 by typing SPEED = 255.

VTAB means vertical tab. It is a statement used to move the cursor to a specific up or down position. VTAB 5 moves the cursor to line 5.

EXAMPLE:

```
1Ø HOME
2Ø PRINT "TYPE YOUR NAME."
3Ø VTAB 15:INPUT N$
4Ø PRINT
5Ø PRINT "YOUR NAME IS ON LINE 15."
```

The number after VTAB must be between 1 and 24. If a number larger than 24 or smaller than 1 is used, an illegal quantity error will result.

MATH MATTERS

The computer works like a calculator, too. It can add, subtract, multiply, and divide.

Here are the symbols the computer uses:

Addition	+
Subtraction	–
Multiplication	*
Division	/

Remember, when typing the numerals one(1) and zero(Ø), use the numbers on the top row of the keyboard. Do not type the letter "l" for 1 or the letter "O" for zero. If you do, you will receive a syntax error message.

Try this program:

```
1Ø PRINT 8+1, 8–1, 8*1, 8/1
2Ø END
RUN
```

Record your results. _____

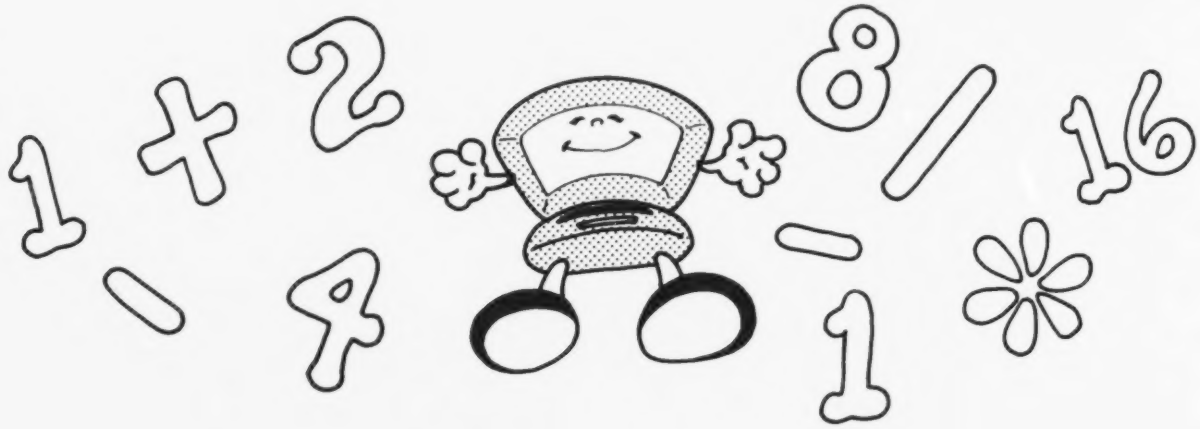
Try this:

```
1Ø PRINT 2*2, 4*4, 8*8, 16*16
2Ø END
RUN
```

Record your results. _____

What happens if you type the above program using semicolons instead of commas?

_____



ARITHMETIC COMPUTATIONS

These symbols are used for arithmetic operations:

+ means "plus" for addition problems.
– means "minus" for subtraction problems.
* means "times" for multiplication problems.
/ means "divided by" for division problems.

Type PRINT 4 + 6 and press RETURN.
The computer responds with 10.

Type PRINT 8 – 7 and press RETURN.
The computer responds with 1.

Type PRINT 4 * 8 and press RETURN.
The computer responds with 32.

Type PRINT 9/3 and press RETURN.
The computer responds with 3.

Suppose you type PRINT 8 + 7 * 3. Will the computer respond with 45 or 29? Try and see. What does this show?

Try these problems. What conclusions can you make?

```
PRINT 6 – 4/2
PRINT 6 * 4/2
PRINT 4 * 3 + 6 * 2
PRINT 2 + 3 * 4
PRINT 8/4 – 3
PRINT 18/(3+1)
PRINT (10/2) + 6
PRINT 3 * 4 + (8–2)
```

As you see, Applesoft BASIC works division and multiplication before addition and subtraction. The operations are done from left to right. The computer works whatever is in parentheses first before solving anything else. If there are parentheses inside parentheses, the computations inside the innermost parentheses are carried out first.

COUNTING PROGRAMS

Here are some counting programs. After studying them and RUNning them, you may want to try writing some of your own.

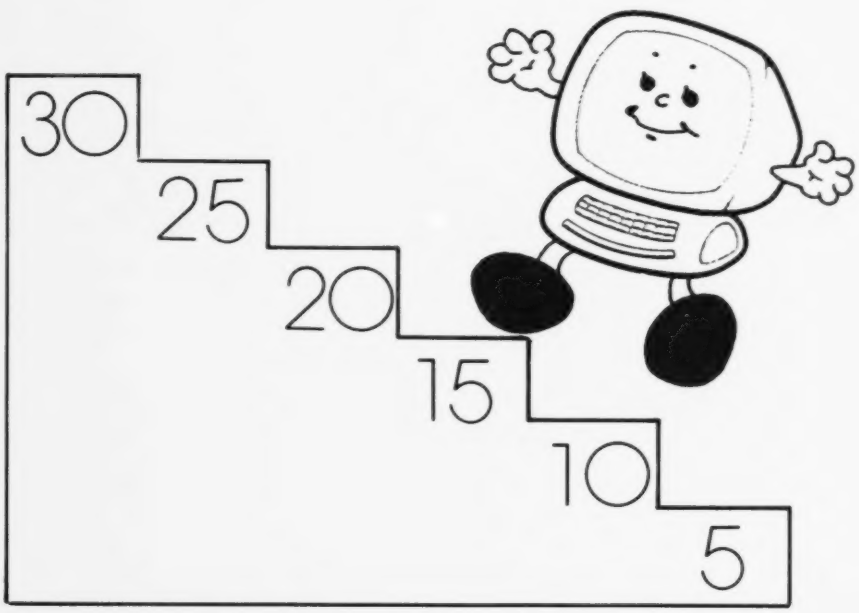
```
NEW
10 REM COUNTING BY FIVE
20 FOR A = 0 TO 1000 STEP 5
30 PRINT A
40 NEXT A
50 END
RUN
```

```
NEW
10 REM COUNTING BY THREE
20 FOR C = 0 TO 999 STEP 3
30 PRINT C
40 NEXT C
50 END
RUN
```

Would you like to see some impressive counting? Change line 20 in the above program to C = 0 TO 10000.

ON YOUR OWN: Write a program that counts by fours.

Write a program that counts by tens.



USING ◄ , ► , AND ◄ ►

You have probably used these three symbols in math.

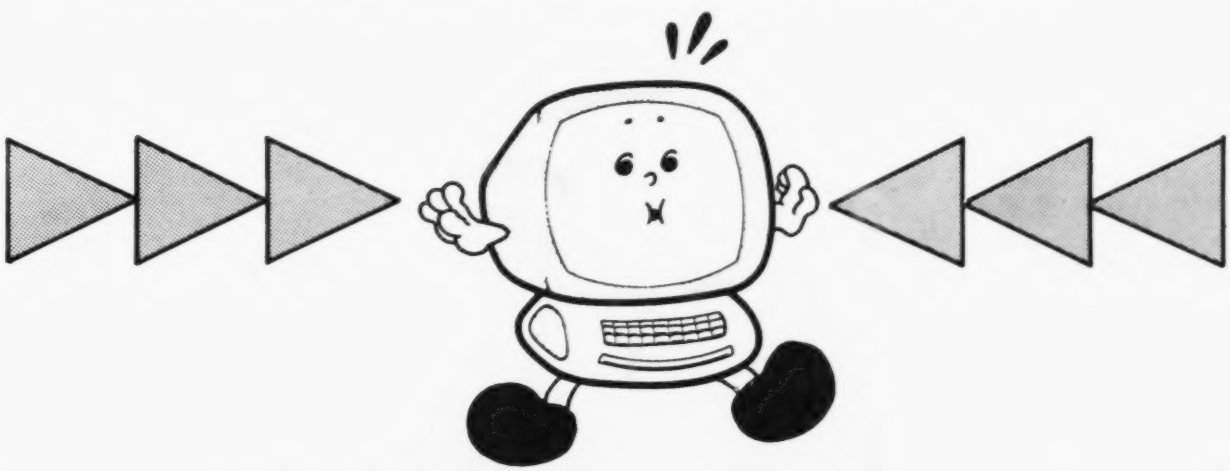
► means "less than."
◄ means "greater than."
◄► means "not equal to."

These symbols are often used in computer programming when decisions must be made. Study the programs below. Then type and RUN them.

```
10 HOME
20 PRINT "TYPE A NUMBER FROM 1 TO 20."
30 INPUT N
40 IF N   0 AND N   10 THEN PRINT "THAT NUMBER HAS ONE DIGIT."
50 IF N   9 AND N   21 THEN PRINT "THAT NUMBER HAS TWO DIGITS."
60 IF N   1 OR N   20 THEN 20
70 END
```

```
NEW
10 HOME
20 PRINT "WHICH DO YOU PREFER, CARS OR TRUCKS?"
30 INPUT C$
40 IF C$ = "CARS" THEN 100
50 IF C$ = "TRUCKS" THEN 130
60 IF C$ ◄► "CARS" AND C$ ◄► "TRUCKS" THEN 10
100 PRINT "ME, TOO. CARS ARE THE BEST THING"
110 PRINT "SINCE CHOCOLATE PIE."
120 END
130 PRINT "I LIKE TRUCKS, TOO, ESPECIALLY"
140 PRINT "ONES LOADED WITH CHOCOLATE PIE."
150 END
```

ON YOUR OWN: Can you write a program that uses ◄ , ► , and ◄ ► symbols?

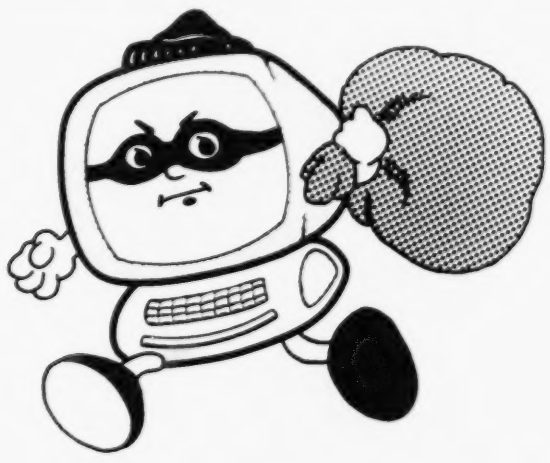


ROBBER CHASE

Can you capture the robber who took the gold from Fort Knox? RUN this program and see.

```
 10 REM ROBBER CHASE
 15 HOME
 20 PRINT "BAD BARNEY BAILEY HAS JUST STOLEN"
 30 PRINT "THE GOLD FROM FORT KNOX."
 40 PRINT "YOU MAY CATCH HIM BY TYPING"
 50 PRINT "A LOCATION NUMBER, 1 - 10."
 60 PRINT "REMEMBER, HE IS ON THE MOVE!"
 65 PRINT: PRINT "TYPE THE NUMBER WHERE YOU THINK HE IS."
 70 G = 10
 75 INPUT N
 77 IF N > 10 THEN PRINT "TYPE A NUMBER, 1 - 10.":GOTO 75
 85 G = G - 1
 90 PRINT
100 T = INT ( RND (1) * 10)
120 IF N > T THEN 200
130 IF N < T THEN 200
140 IF N = T THEN 500
145 GOTO 75
200 PRINT "TOO LATE. YOU JUST MISSED HIM."
210 IF G = 1 THEN 225
215 IF G = 0 THEN 600
220 PRINT "YOU MAY LOOK ";G;" MORE PLACES."
225 IF G = 1 THEN PRINT "YOU MAY LOOK ONE MORE PLACE."
230 IF G = 0 THEN 600
250 GOTO 75
500 PRINT "WOWEE! YOU CAUGHT THE ROBBER AND SAVED FORT KNOX!"
550 END
600 PRINT "THE ROBBER GOT AWAY! CALL THE FBI."
650 END
```

(space)

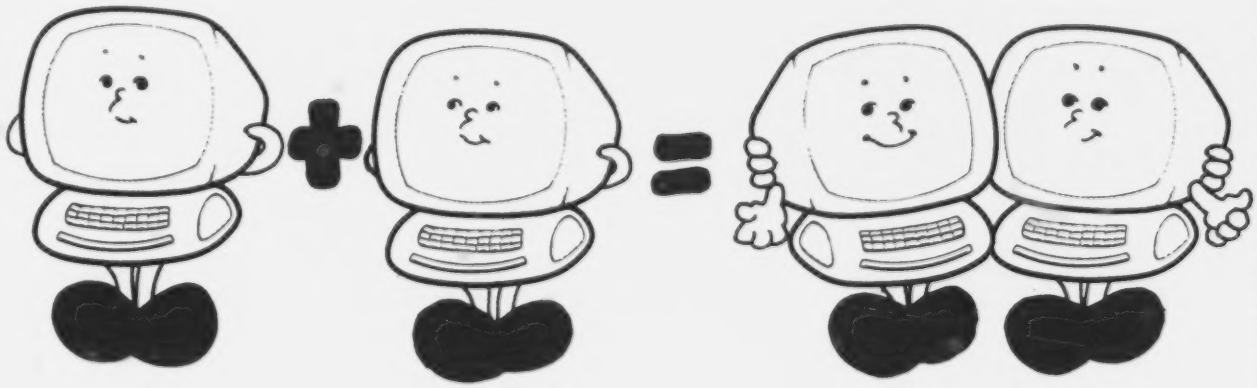


ADDITION, SUBTRACTION, AND MULTIPLICATION

```
10 REM ADDITION, SUBTRACTION, AND MULTIPLICATION OF TWO
   NUMBERS
15 HOME
20 PRINT "TYPE A NUMBER."
25 INPUT A
30 PRINT "TYPE ANOTHER NUMBER."
35 INPUT B
40 LET S=A+B
45 LET D=A-B
50 LET P=A*B        (space)
60 PRINT "THE SUM IS ";S"."(space)
70 PRINT "THE DIFFERENCE IS ";D"."
80 PRINT "THE PRODUCT IS ";P"."
100 PRINT: GOTO 20          (space)
```

ON YOUR OWN: Write a program that will add three numbers.

Write a program that will add five numbers to the product of any two numbers.



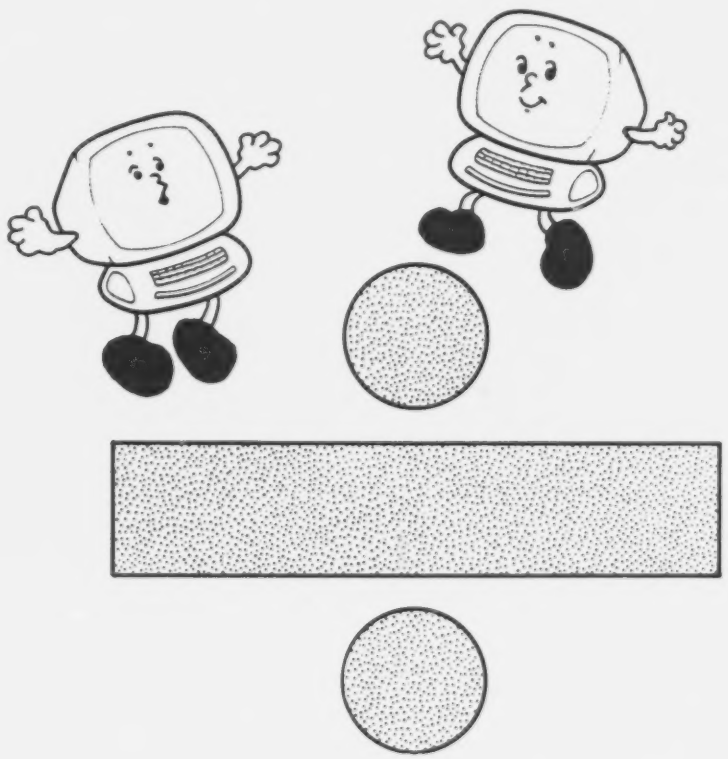
LONG DIVISION

Here is a program that will do long division.

```
10 REM LONG DIVISION
20 HOME
30 REM X = DIVIDEND: Y = DIVISOR: R = REMAINDER: Q = QUOTIENT
40 PRINT "TYPE THE DIVIDEND."; (space)
50 INPUT X: INPUT "DIVIDED BY ";Y
60 Q = INT(X/Y)
70 R = X - Y*Q                          (space)
80 PRINT:PRINT "THE QUOTIENT IS ";Q".": PRINT "THE REMAINDER
   IS ";R"." (space)
90 PRINT
100 GOTO 40
```

Look at line 60 in the program. INT stands for integer. INT(X/Y) tells the computer to find the whole number (without the remainder) when the dividend is divided by the divisor.

When you type the program, be sure to put a space after the word IS in line 80. That will make the spacing correct when the answer is printed.



AVERAGING YOUR GRADES

This program will help you average your grades in any subject.

```
 10 HOME
 20 PRINT "HOW MANY GRADES DO YOU HAVE?"
 30 INPUT T
 40 PRINT "TYPE YOUR GRADES, ONE GRADE PER LINE."
 50 LET S=0
 60 FOR I=1 TO T
 70 INPUT X
 80 LET S=S+X
 90 NEXT I
100 LET A=S/T
110 PRINT                                (space)
120 PRINT "YOUR GRADE AVERAGE IS ";A"."
130 END
```

Can you answer the following questions about the program?

1. Which line number tells the computer the number of grades that are to be averaged?

   _____

2. Which line number requests the first grade?

   _____

3. Which two line numbers are necessary in order to compute the average?

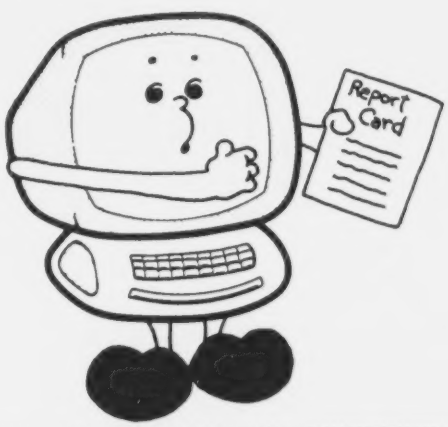
   _____

4. Which line number displays the average?

   _____

5. What is the purpose of line 10?

   _____



LOVE LIFE FORECAST

```
 1Ø REM LOVE LIFE FORECAST
 2Ø HOME
 3Ø PRINT "WHAT IS YOUR NAME?";N$
 4Ø INPUT N$ (space)
 5Ø PRINT N$;",I AM THE LOVE LIFE GENIE."
 6Ø PRINT "I HAVE A FORECAST FOR YOU."
 7Ø PRINT "FIRST, YOU MUST TYPE THE NAME OF YOUR LOVE,"
 8Ø PRINT "AND I WILL TELL YOU MY FORECAST."
 9Ø PRINT
1ØØ INPUT L$
11Ø IF LEN(L$) =2 THEN 21Ø
12Ø IF LEN(L$) =3 THEN 23Ø
13Ø IF LEN(L$) =4 THEN 25Ø
14Ø IF LEN(L$) =5 THEN 27Ø
15Ø IF LEN(L$) =6 THEN 29Ø
16Ø IF LEN(L$) =7 THEN 31Ø
17Ø IF LEN(L$) =8 THEN 33Ø
18Ø IF LEN(L$) =9 THEN 35Ø
19Ø IF LEN(L$) = 1Ø THEN 37Ø
2ØØ IF LEN(L$) ◄ 2 OR LEN(L$) ► 1Ø THEN 39Ø
21Ø PRINT "WATCHOUT! I SENSE HEARTBREAK!"
22Ø END
23Ø PRINT "YOUR LOVE TURNS THE HEADS!"
24Ø END
25Ø PRINT "OOOH! A SWEET TALKER!"
26Ø END
27Ø PRINT "YOUR LOVE LIKES A BIG SPENDER."
28Ø END
29Ø PRINT "WHAT A CHARMER!"
3ØØ END
31Ø PRINT "BEWARE! YOUR LOVE COULD BE A TWO-TIMER."
32Ø END
33Ø PRINT "I THINK I HEAR WEDDING BELLS!"
34Ø END
35Ø PRINT "I SENSE TRUE LOVE."
36Ø END
37Ø PRINT "I SEE A HEART OF LOVE."
38Ø END
39Ø PRINT "GET READY FOR THE WEDDING!"
4ØØ END
```

WINNINGS!

Lucky you! You just won a sweepstakes prize of $200 a month for ten years. RUN the program below to see how much your prize is.

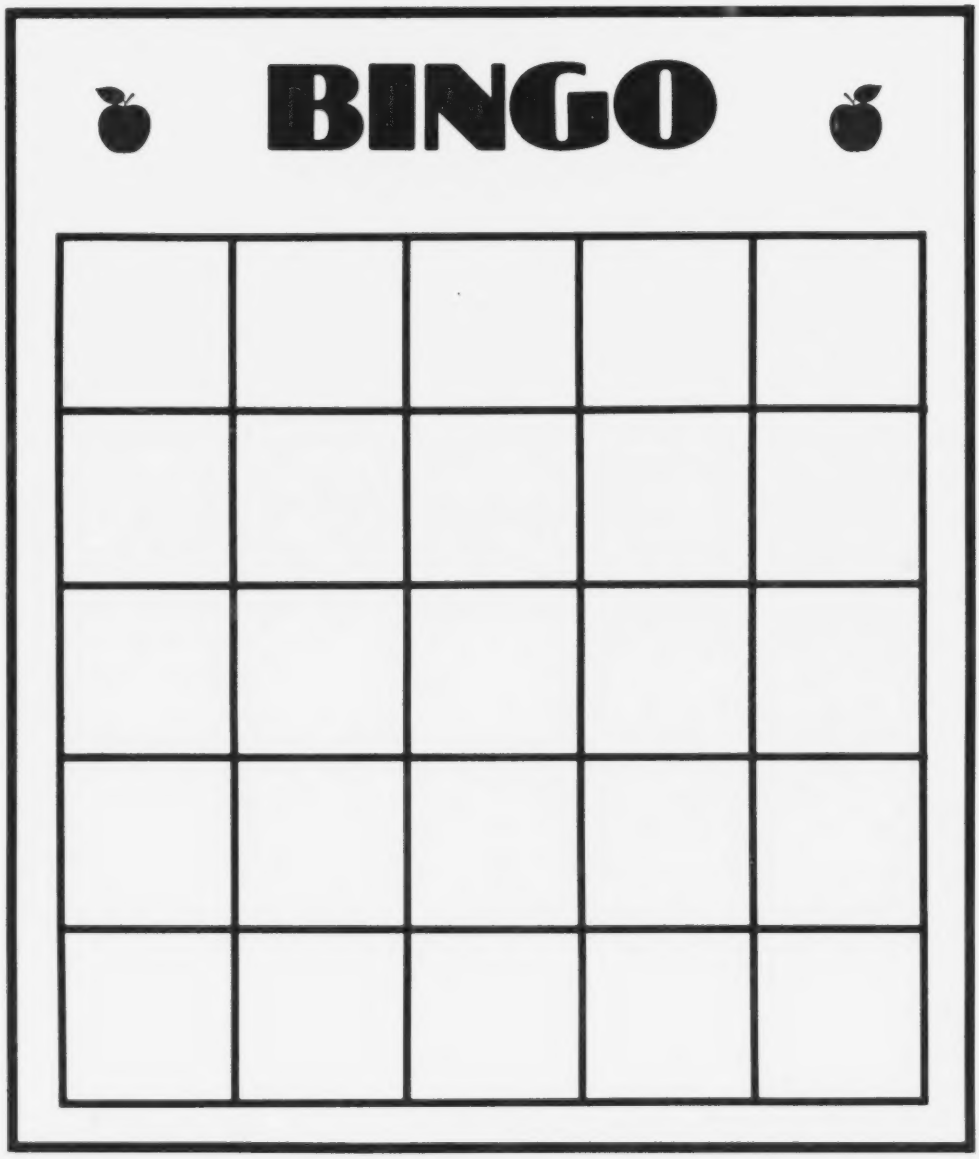
```
10 REM WINNINGS
20 HOME
30 A = 200
40 B = 10
50 C = 12
60 D = (A * C) * B
70 PRINT "CONGRATULATIONS!"
80 PRINT "YOUR PRIZE MONEY TOTALS ";D;" DOLLARS."
90 END
```

(space)

ON YOUR OWN: Write a program that computes a different amount of money for a different time period.

BASIC BINGO

Now you can play Bingo with the computer as the caller. First, you must make your playing card. Just fill in the spaces on the blank card below with any numerals from Ø to 5Ø. The placement of the numerals does not matter. Make the card to suit yourself.



Type the BASIC Bingo program that is listed on the next page. You may want to save it on a disk or cassette tape for later use.

BASIC BINGO

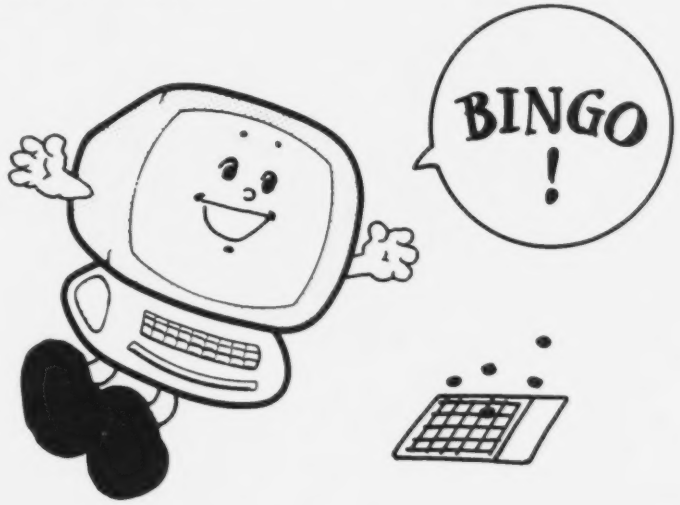
Type this program to use with your Bingo card. The monitor will display the numbers when you are ready to play.

```
 10 REM BASIC BINGO
 20 HOME
 30 PRINT "LET'S PLAY BASIC BINGO."
 40 PRINT
 50 PRINT "I AM YOUR CALLER."
 60 PRINT "I WILL CALL NUMBERS AT RANDOM."
 70 PRINT "WHEN I CALL A NUMBER THAT IS ON YOUR"
 80 PRINT "CARD, COVER THAT SPACE WITH A MARKER."
 90 PRINT "YOU WIN WHEN YOU COVER AN ENTIRE ROW"
100 PRINT "ACROSS, UP AND DOWN, OR DIAGONALLY."
110 PRINT: PRINT "PRESS ANY KEY WHEN YOU ARE READY."
120 GET K$: HOME
130 PRINT "HERE WE GO."
140 PRINT
150 X = INT(50*RND(1) )
160 PRINT X
170 PRINT
180 FOR I = 1 TO 2500:NEXT I
190 GOTO 150
```

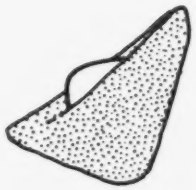
Line 120 contains a GET K$ which puts a blinking cursor on the screen. It waits for the person at the keyboard to strike a key before going further.

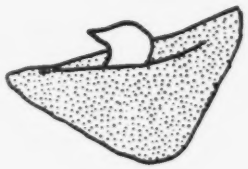
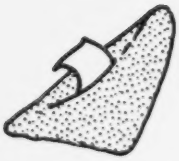
You may change the 2500 in line 180 to a larger number to slow the calling speed of the numbers. A smaller number will speed up the time between numbers.

Press CONTROL-C to interrupt the numbers when a winning card is found. Then type RUN and press RETURN to play again.



FORTUNE COOKIES



```
 10 REM FORTUNE COOKIES
 20 HOME
 30 PRINT "HELP YOURSELF TO A FORTUNE COOKIE."
 40 PRINT "SELECT A COOKIE BY TYPING A NUMBER 1 TO 12."
 50 INPUT C
 60 PRINT
 70 IF C = 1 THEN 200
 80 IF C = 2 THEN 230
 90 IF C = 3 THEN 260
100 IF C = 4 THEN 290
110 IF C = 5 THEN 310
120 IF C = 6 THEN 330
130 IF C = 7 THEN 350
140 IF C = 8 THEN 370
150 IF C = 9 THEN 390
160 IF C = 10 THEN 410
170 IF C = 11 THEN 430
180 IF C = 12 THEN 450
190 IF C < 1 OR C > 12 THEN 40
200 PRINT "A SURPRISE AWAITS YOU!"
210 PRINT "YOU MAY HAVE ANOTHER COOKIE."
220 GOTO 40
230 PRINT "YOU WILL RECEIVE SOME GOOD NEWS!"
240 PRINT "PLEASE HAVE ANOTHER COOKIE."
250 GOTO 40
260 PRINT "SMILE! IT IS YOUR DAY."
270 PRINT "HAVE ANOTHER COOKIE."
280 GOTO 40
290 PRINT "GOOD THINGS AWAIT YOU!"
300 GOTO 210
310 PRINT "BEWARE OF STRANGE DOGS."
320 GOTO 240
330 PRINT "SAVE SOME MONEY FOR A RAINY DAY."
340 GOTO 270
350 PRINT "BE KIND TO ANIMALS!"
360 GOTO 210
370 PRINT "DO A KIND DEED FOR SOMEONE."
380 GOTO 240
390 PRINT "YOU CAN BRIGHTEN SOMEONE'S DAY."
400 GOTO 270
410 PRINT "GUARD YOUR HEALTH. IT IS PRECIOUS."
420 GOTO 210
430 PRINT "OTHERS MAY FOLLOW YOUR EXAMPLE."
440 GOTO 240
450 PRINT "KEEP YOUR TEMPER. NO ONE WANTS IT."
460 GOTO 270
470 END
RUN
```

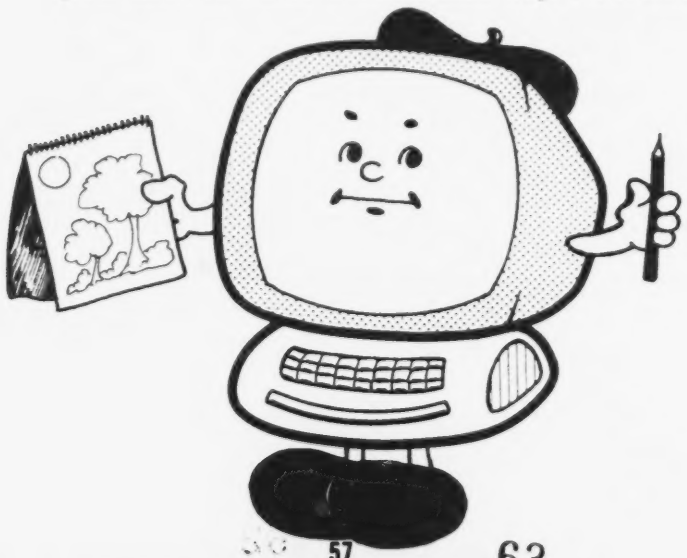
DRAWING WITH THE KEYBOARD

Simple drawings can be made using the symbols, letters, and numbers on the keyboard. PRINT statements tell the computer what to draw. The symbols that are used in the drawings must be enclosed inside quotation marks. Below are two ideas to get you started.

```
10 HOME
20 PRINT "    *******"
30 PRINT "   *********"
40 PRINT "  ***********"
50 PRINT " ************"
60 PRINT "  ***********"
70 PRINT "   *********"
80 PRINT "      ###"
90 PRINT "      ###"
100 PRINT "     ###"
110 PRINT "     ###"
120 PRINT "     ###"
130 PRINT "   =###="
140 END
```

```
NEW
10 HOME
20 PRINT "[ ] [ ] [ ] [ ] [ ] [ ]"
30 PRINT "[ ]                 [ ]"
40 PRINT "[ ]   O     O     [ ]"
50 PRINT "[ ]      [ ]        [ ]"
60 PRINT "[ ]    -------      [ ]"
70 PRINT "[ ]                 [ ]"
80 PRINT "[ ] [ ] [ ] [ ] [ ] [ ]"
90 END
```

ON YOUR OWN: See what you can draw using the keyboard. You might like to write your name or initials in an unusual way. Have fun!



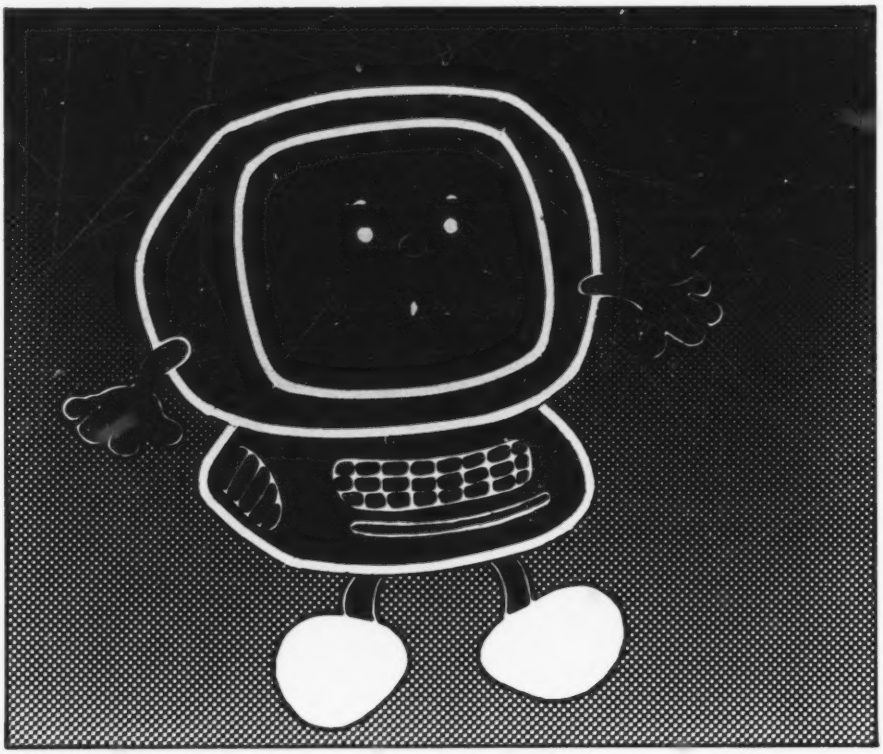
INVERSE

The INVERSE statement is used to print black text on a white background. Most programmers use this technique in very small doses for special effects. The NORMAL statement is used to return from INVERSE to the regular screen.

EXAMPLE:

```
10 HOME
20 PRINT "TO CREATE A SPECIAL IMPRESSION"
30 PRINT "SOME PROGRAMMERS USE"
40 INVERSE
50 PRINT "AN INVERSE STATEMENT."
60 NORMAL
```

ON YOUR OWN: Experiment with the INVERSE statement. Remember to use the NORMAL statement (as in line 60) to return the screen to its regular dark background.



FLASH

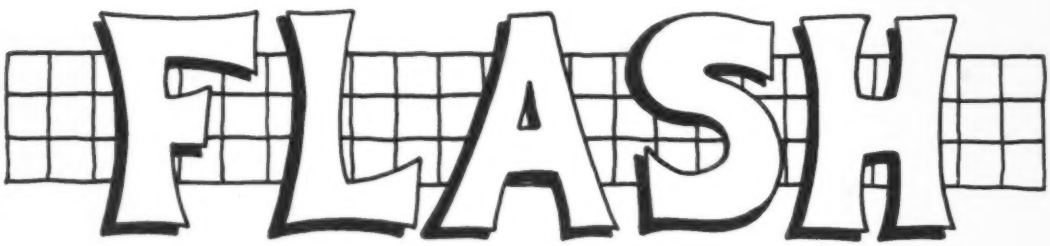
The FLASH statement is used to flash words, letters, numbers, or other symbols on the screen. It alternately displays text between black-on-white and white-on-black. Most programmers use FLASH only for highlighting important messages. It is stopped with the NORMAL statement.

Try this program:

```
 10 HOME
 20 PRINT "WHAT IS THE CAPITAL OF ALABAMA?"
 30 INPUT C$
 40 IF C$ = "MONTGOMERY" THEN 100
 50 IF C$ <> "MONTGOMERY" THEN 200
100 FLASH
110 PRINT "CORRECT!"
120 NORMAL
125 END
200 FLASH
210 PRINT "INCORRECT!"
220 NORMAL
230 END
```

Here is another program to demonstrate the use of FLASH.

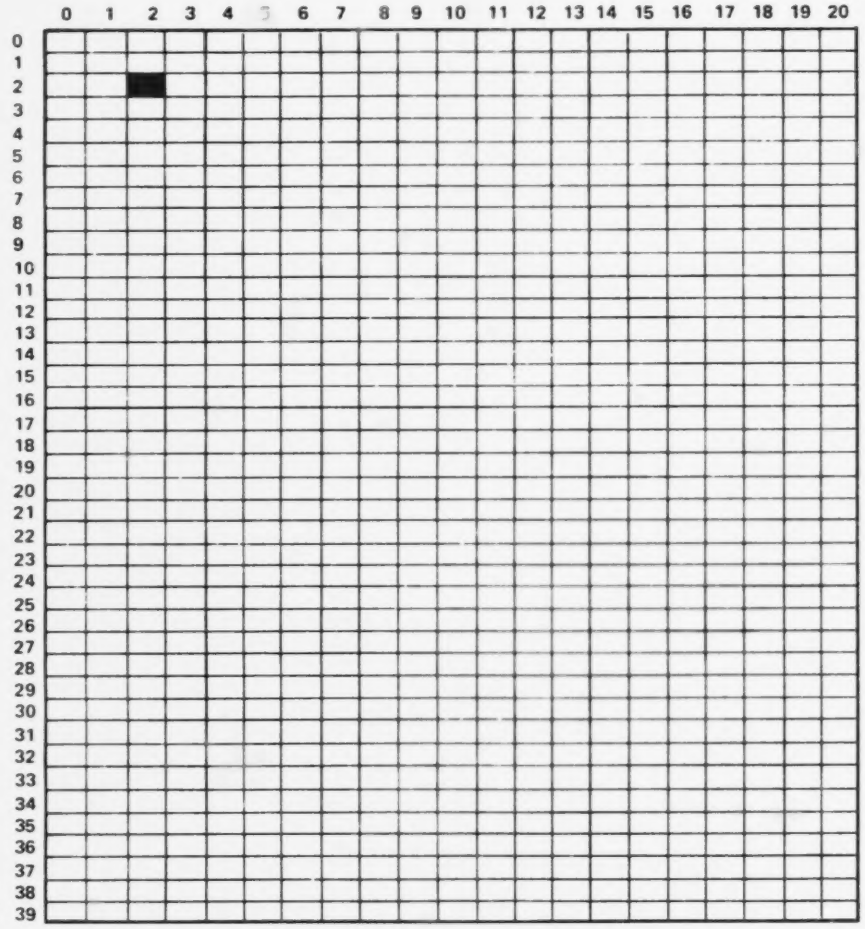
```
NEW
 10 HOME
 20 PRINT "WHAT IS YOUR FAVORITE COLOR?"
 30 INPUT C$
 40 HOME
 50 FLASH
 60 HTAB 12
 70 VTAB 12
 80 PRINT C$
 90 NORMAL
RUN
```



GRAPHICS

So far, our programs have been in the text mode. We can also use BASIC in the graphics mode. We use graphics to draw shapes and pictures.

Imagine your video display screen as a sheet of graph paper. The rows and columns of graph paper form a grid. The horizontal rows are referred to as X. The vertical columns are referred to as Y.



A position can be located on the grid with an X,Y point. The block that is shaded is 2,2.

Shade in 6,1. GO ACROSS and THEN DOWN.

LOW-RESOLUTION GRAPHICS

Turn on your computer and bring up *Applesoft BASIC. Now type GR. GR stands for graphics. Press RETURN. What happens to the display screen when you type GR and press RETURN?

The graphics grid in low-resolution graphics has 40 blocks across and 40 blocks down. That is the type grid we will be using. The blocks are rectangular and are numbered from 0 to 39.

Try this:

```
10 GR
20 COLOR = 6
30 PLOT 10,15
40 PLOT 15,10
RUN
```

Line 10 enters the graphics mode.
Line 20 sets the color to medium blue.
Line 30 goes across to 10 and down to 15 to color a block blue.
Line 40 goes across to 15 and down to 10 to color a block blue.

If you do not have a color monitor, you will not see a blue block, but you can still enjoy differences of shading.

ON YOUR OWN: Experiment with PLOT. Use it to shade in whatever blocks you like. Remember to think of a 40 x 40 grid. Go across and down to locate a block. Separate each pair of numbers with a comma. Do not use a number larger than 39.

NOTE: Type TEXT and press RETURN to get out of graphics. Type HOME and press RETURN to clear the screen. The screen will then be ready to display text letters typed from the keyboard.

*Apple and Applesoft are trademarks of Apple Computer, Inc.

PLOTTING LINES

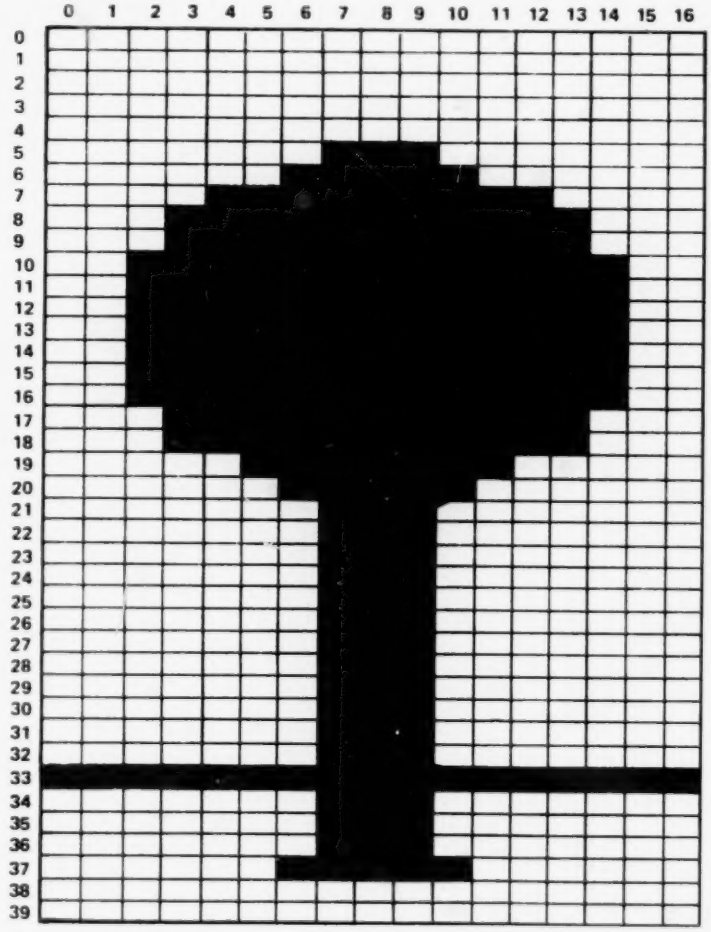
The HLIN statement is used to draw a horizontal line in *Applesoft. VLIN is used to draw a vertical line.

EXAMPLES: HLIN 2,8 AT 4 draws a line beginning at column 2 and ending at column 8 on the fourth row.

VLIN 6,12 AT 2Ø draws a vertical line at column 2Ø beginning at row 6 and ending at row 12.

Type and RUN this program:

```
 5 GR
1Ø COLOR = 12
2Ø HLIN 7,9 AT 5
3Ø HLIN 6,1Ø AT 6
4Ø HLIN 4,12 AT 7
5Ø HLIN 3,13 AT 8
6Ø HLIN 3,13 AT 9
7Ø HLIN 2,14 AT 1Ø
8Ø HLIN 2,14 AT 11
9Ø HLIN 2,14 AT 12
1ØØ HLIN 2,14 AT 13
11Ø HLIN 2,14 AT 14
12Ø HLIN 2,14 AT 15
13Ø HLIN 2,14 AT 16
14Ø HLIN 3,13 AT 17
15Ø HLIN 3,13 AT 18
16Ø HLIN 5,11 AT 19
17Ø HLIN 6,1Ø AT 2Ø
18Ø HLIN 7,9 AT 21
19Ø COLOR = 8
2ØØ VLIN 22,37 AT 7
21Ø VLIN 22,37 AT 8
22Ø VLIN 22,37 AT 9
23Ø PLOT 6,37
24Ø PLOT 1Ø,37
25Ø HLIN Ø,39 AT 33
```



*Applesoft is a trademark of Apple Computer, Inc.

CREATING WITH GRAPHICS

Now you are ready to enjoy the computer as a drawing screen. Think of all the things you can draw! The possibilities are endless.

Use a blank grid like the one on the following page to plan your drawing. Shade in the blocks to create a picture. Then locate the blocks and lines that were used in your drawing. Remember, each location must have two numbers when you are plotting a block and three numbers when you are making a line. Locate your position by first counting across the grid and then counting down.

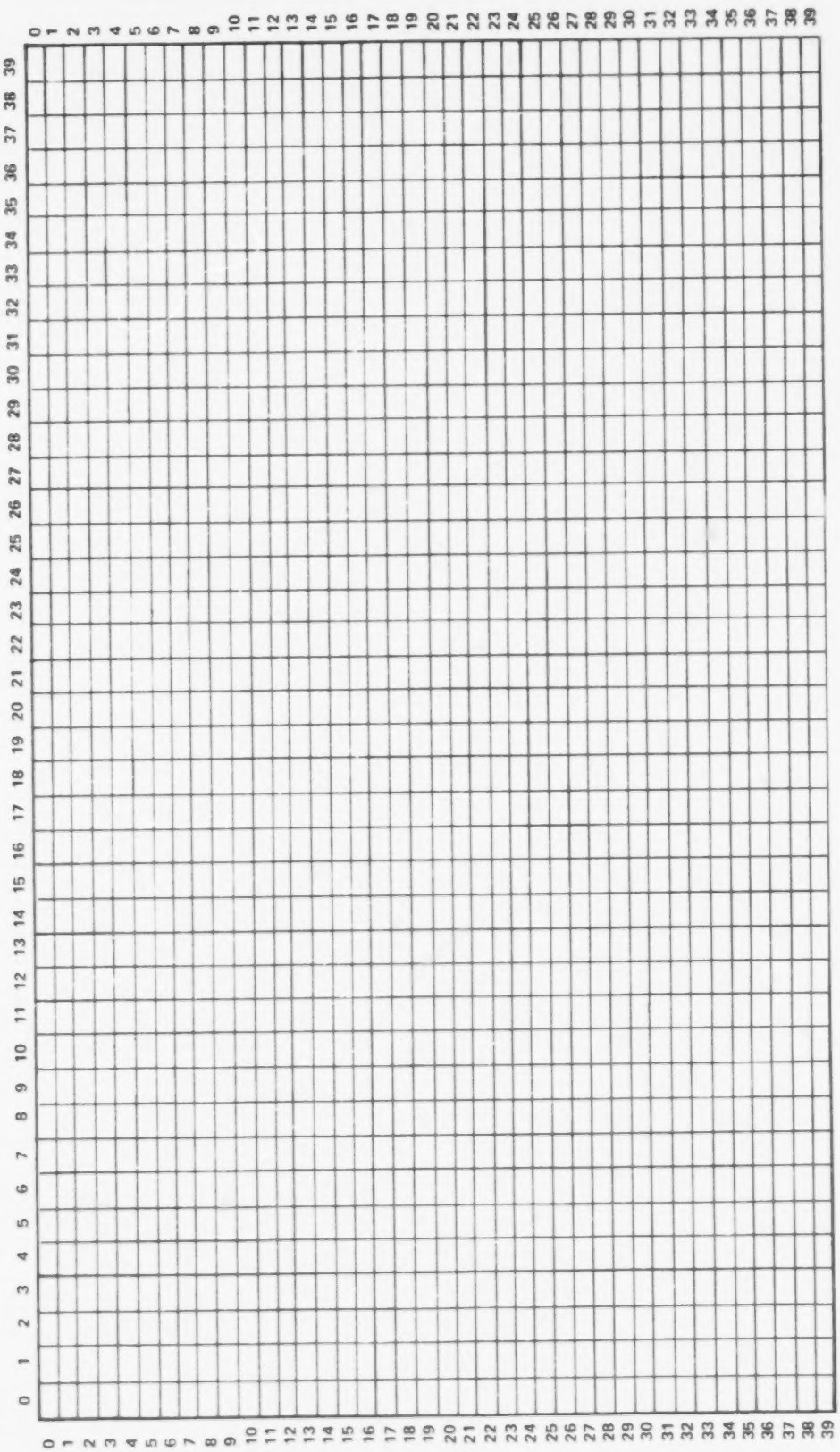
PLOT is used to color a single block.
(Example: PLOT 6,7)

HLIN is used to produce a horizontal line.
(Example: HLIN 3,8 AT 2Ø)

VLIN is used to produce a vertical line.
(Example: VLIN 4,2Ø AT 8)

Be sure to type GR in the first line number of a graphics program. Use a line number before each PLOT, HLIN, and VLIN statement.

A BLANK GRID



HIGH-RESOLUTION GRAPHICS

High-resolution graphics allows for individual control of each dot on the screen. Pictures can be drawn to show more detail than low-res because the dots are seven times smaller than the low-res blocks. However, hi-res does take up much more computer memory.

High-resolution (hi-res) graphics on the Apple computer has six colors. There are 280 points across the screen and 192 points down the screen that can be accessed in hi-res graphics.

AVAILABLE COLORS:

0 BLACK
1 GREEN
2 VIOLET
3 WHITE
4 BLACK
5 ORANGE
6 BLUE
7 WHITE

Notice, there are two whites and two blacks.

HCOLOR is used to select the color in hi-res graphics.

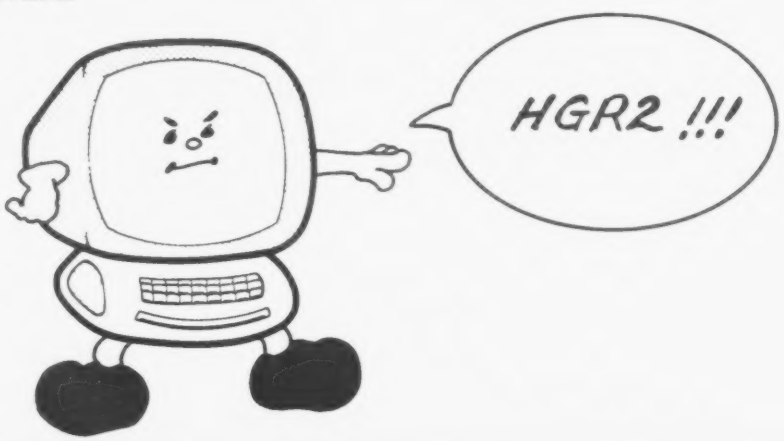
EXAMPLE: 30 HCOLOR = 6 (This sets the color to blue.)

The command HGR is used to tell the computer we would like to use page 1 memory of hi-res graphics. The command is used with a line number followed by a space.

EXAMPLE: 10 HGR

HGR2 is a command used to tell the computer we would like to use page 2 of hi-res graphics. It is used with a line number and a space.

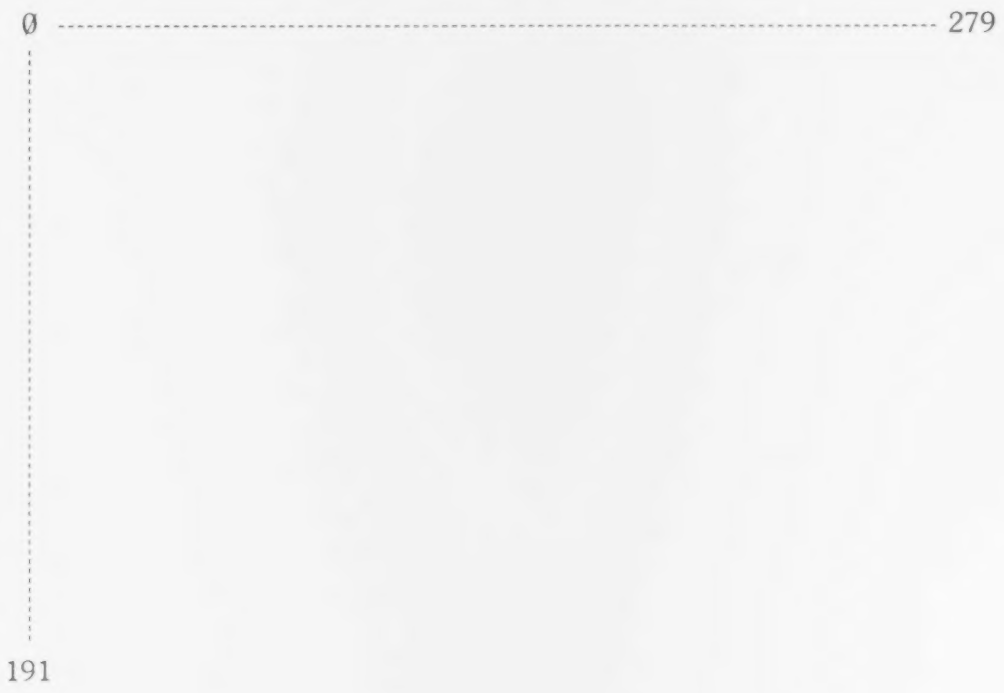
EXAMPLE: 10 HGR2



HIGH-RESOLUTION GRAPHICS

There are 280 points across the hi-res screen. They are numbered from 0 to 279. Each point is called a pixel.

There are 192 points down the screen. The points are numbered from 0 to 191.



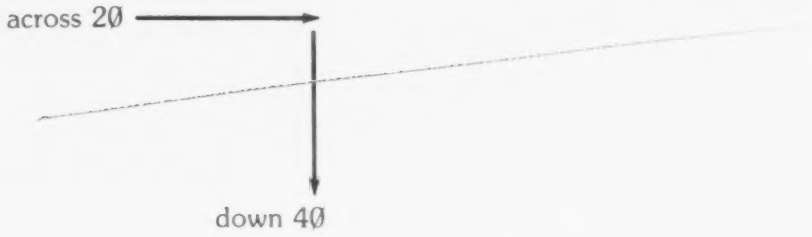
Here is a program that randomly plots blue pixels on the hi-res screen.

```
10 HGR
20 HCOLOR = 6
30 FOR I = 1 TO 500
40 X = RND (1) * 279
50 Y = RND (1) * 191
60 HPLOT X,Y
70 NEXT I
```

HI-RES GRAPHICS

Each dot or pixel on the hi-res screen can be turned on or off. HPLOT is used to turn on a dot. Its location is given by counting across first (the X number) and then down (the Y number).

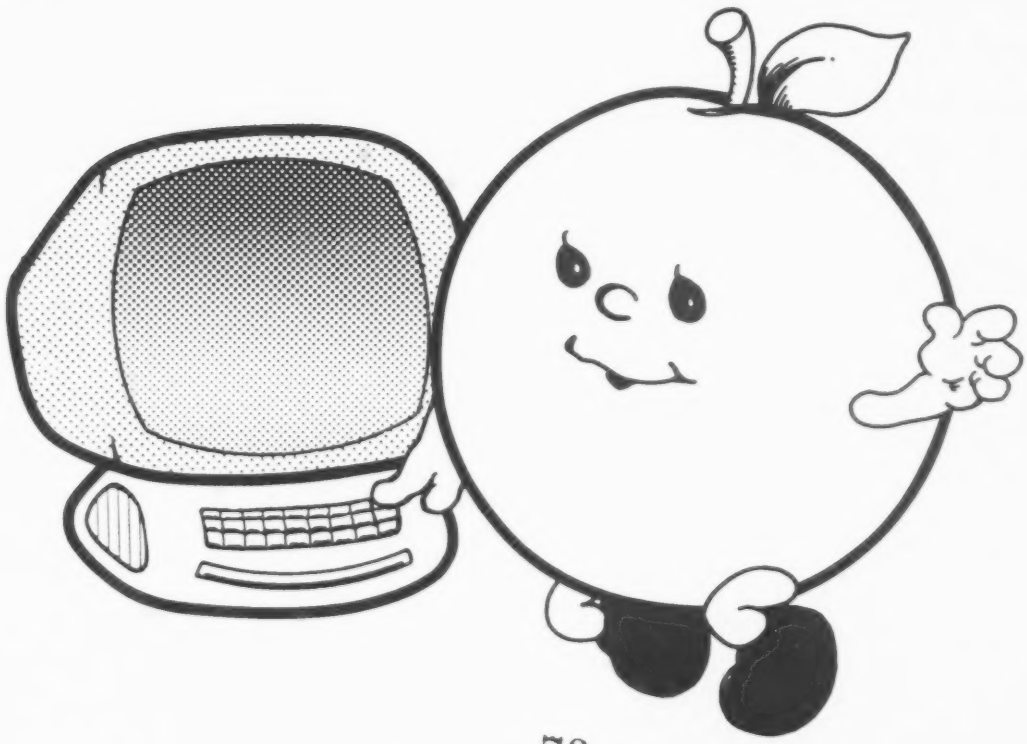
EXAMPLE: HPLOT 20, 40



Here is a program that goes across 20 and down 40 to plot a violet dot. You may like to add additional line numbers to plot other pixels.

```
10 HGR
20 HCOLOR = 2
30 HPLOT 20, 40
RUN
```

ON YOUR OWN: Practice turning pixels on and off. Remember, 279 is the largest number across, and 191 is the largest number going down.



DRAWING WITH HPLOT

HPLOT means high-resolution plot. It is used to draw points and lines in high-resolution graphics.

EXAMPLE:

```
10 HGR
20 HCOLOR = 6
30 HPLOT 80,40
RUN
```

A blue dot is drawn at column 80, row 40.

A line can be drawn with HPLOT by naming the beginning and ending points, separated by the keyword TO.

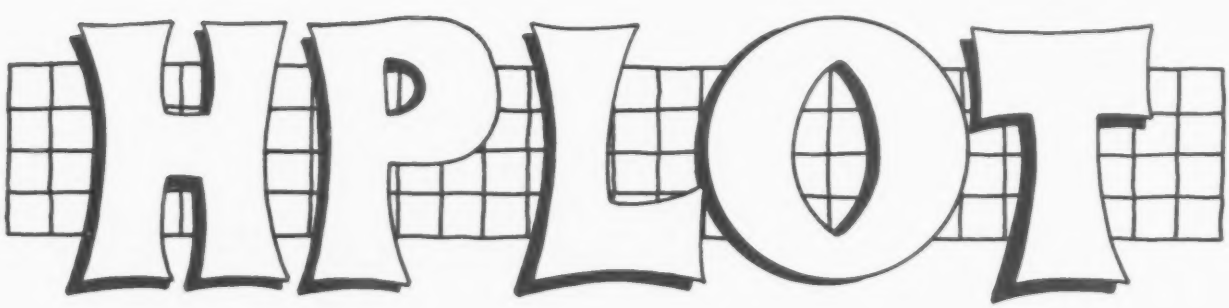
EXAMPLE:

```
10 HGR
20 HCOLOR = 6
30 HPLOT 0, 40 TO 200, 40
RUN
```

The program below draws a rectangle in blue.

```
10 HGR
20 HCOLOR = 6
30 HPLOT 60, 30 TO 200, 30 TO 200, 110 TO 60, 110 TO 60, 30
```

NOTE: The HPLOT range extends from 0 to 279. Any number larger than 279 will cause an illegal quantity error.



HI-RES GRAPHICS

Information is stored in different addresses in the computer's memory. High-resolution graphics has two areas of memory to use. HGR, or hi-res page 1, has addresses 8192 through 16383. HGR2, or hi-res page 2, has addresses 16384 through 24575.

When you type HGR, you are turning off all the bits in page 1 of hi-res memory. This is why the screen appears black. A bit is the smallest unit of memory in a computer.

Different points or pixels can be turned on by using HPLOT to pinpoint different locations.

Since the addresses for hi-res page 1 are 8192 through 16383, we can turn on each pixel one by one with the program below.

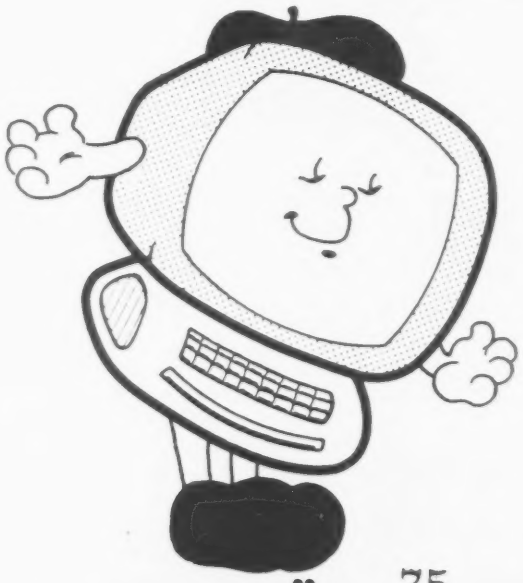
```
10 HGR
20 FOR P = 8192 TO 16383
30 POKE P,1
40 NEXT P
```

Do you understand how the program works? Line 10 changes the screen to hi-res page 1. Line 20 sets up the loop that will be repeated until all the pixels are turned on. Line 30 puts the first value into the first address, 8192. Line 40 increases the value by 1 each time until 16383 is reached.

You can plot every other dot by changing Line 20. Type 20 FOR P = 8192 TO 16383 STEP 2.

If you would like to see the pixels turned on from the bottom of the screen upward, change Line 20 again. Type 20 FOR P = 16383 TO 8192 STEP −1.

If you are interested in drawing hi-res pictures, there are many books on the market to help you go further. There also are software packages and graphics tablets that make the drawing easier.



WHICH IS CORRECT?

Directions: Study each entry. If it is correct, place a check mark under the column labeled CORRECT. If the entry contains an error, place a check mark under the INCORRECT column.

	CORRECT	INCORRECT
1. 1Ø PRINT "HELLO		
2. 1Ø "PRINT HELLO"		
3. 1Ø PRINT "HELLO"		
4. 2Ø GOTO "HELLO"		
5. 3Ø GOTO 1Ø		
6. 6Ø LIST "PROGRAM"		
7. 4Ø PRINT NEW		
8. LIST		
9. IF A◄B PRINT		
10. LET C="CAT"		
11. PRINT ADDRESS		
12. 1Ø PRINT REM		
13. 1Ø REM SPELLING GRADES		
14. PRINT 2+2=4		
15. PRINT 3–1		
16. PRINT N$		
17. GOTO END		
18. 1Ø INPUT N		
19. 5Ø ? "SMILE"		
20. 1ØØ PRINT		

QUESTIONS OFTEN ASKED ABOUT BASIC

1. When you are typing a program, why doesn't the computer carry out the instructions when it completes each line?—The instructions are stored in the computer's memory until the RUN command is typed. RUN instructs the computer to carry out the program.

2. How can you make the computer place a blank line in a program?—Type PRINT and press RETURN. This makes a blank line. It is sometimes called a line feed. You may also type a question mark (?) and press RETURN. Both PRINT and the question mark must be used with a line number.

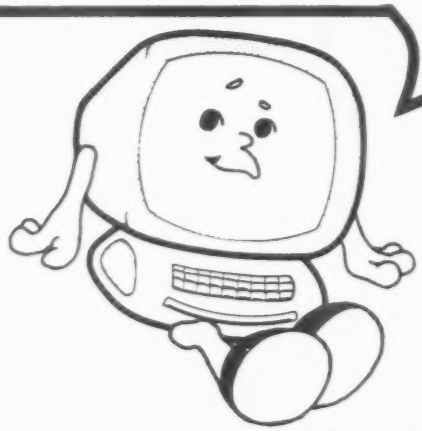
3. Why isn't the REM printed in a program?—REM stands for remark. It is information for the programmer only. It is included to make the reading of the program instructions easier.

4. Do you have to type the lines of a program in the order that they are to be RUN?—It is easier to type a program in line number order, but the computer will run the program in line number order anyway, not in the order the lines are typed. You may type the lines out of line number order, and the computer still reads them in line number order, unless directed to a different line number.

5. Why are programs written with gaps between line numbers instead of 1, 2, 3 order?—That is so there will be room to insert extra lines in a program if necessary. The lines will not have to be retyped or renumbered. Just add a line number that comes between the numbers where the program line belongs.

6. What do you do when you no longer want a program and want to type a new one?—Type NEW and press RETURN. That clears the memory of the computer. Then type the next program.

7. How can a line in a program be replaced with another line?—Type the line number of the statement to be replaced. Then type the new line following the line number. Press RETURN. The old line will be replaced with the new line. You may LIST the program to be sure.



QUESTIONS OFTEN ASKED ABOUT BASIC (cont'd.)

8. Why is it necessary to press RETURN after each line?—Pressing RETURN is your way of telling the computer you have finished your instruction. The computer does not respond until it has been told the line is complete. RETURN tells it that.

9. What happens to a program that is on the screen when the computer is turned off?—The program will be lost. It will have to be typed again if you do not have a copy of it on tape or disk.

10. Is it necessary to type END as the last line in a program?—END can be left out if you wish. That will save space in the memory of the computer. The word END helps clearly define the program's conclusion.

NOTES

COMPUTER QUIZ #1

Directions: Fill in each blank with a word that will make each sentence correct.

1. To erase a program from the computer's memory, type _____
   and press _____.
2. To view the instructions of a program, type _____
   and press _____.
3. To carry out a program, type _____ and press _____.
4. Two ways to instruct the computer to print are _____
   and _____.

Directions: Write C for command and K for keyword in each blank.

_____ 1. LET

_____ 2. RUN

_____ 3. REM

_____ 4. PRINT

_____ 5. END

_____ 6. NEW

_____ 7. GOTO

_____ 8. LIST

_____ 9. NEXT

_____ 10. INPUT

Directions: Match the words with their definitions. Put the letter of the definition in the blank by the correct word.

_____	1. program	A. Repeats a section of a program
_____	2. RUN	B. Clears the computer's memory
_____	3. LIST	C. Carries out a program
_____	4. loop	D. A list of numbered instructions used by a computer
_____	5. REM	E. Displays program instructions
_____	6. string	F. A line written to make a program easier to read
_____	7. variable	G. Any characters inside quotation marks
_____	8. NEW	H. A storage space in the computer's memory

COMPUTER QUIZ #2

Directions: Write what you think the computer will print if each entry is typed and RUN.

1. 1Ø PRINT 2 + 3

   _____

2. 1Ø PRINT "2 + 3"

   _____

3. 1Ø PRINT "HELLO"

   _____

4. 1Ø INPUT N

   _____

5. 1Ø

   _____

6. 1Ø ? "GOOD MORNING"

   _____

7. 1Ø PRINT 3 * 4

   _____

8. 1Ø PRINT 8 / 2

   _____

9. 1Ø REM SPELLING GRADES

   _____

10. 1Ø PRINT "THE SUM OF 2 AND 6"

    _____

Directions: Write a short program that will find the sum of two numbers.

COMPUTER QUIZ #3

Directions: Circle the letter of the word or phrase that correctly completes each statement.

1. A computer's display of pictures on the screen is called
   a. computer printout
   b. computer graphics
   c. text display

2. The BASIC command to remove a program from RAM is
   a. RESET
   b. NEW
   c. ERASE

3. RUN is a BASIC
   a. command
   b. statement
   c. code

4. The video screen display is called
   a. output
   b. memory
   c. graphics

5. A set of computer instructions is called
   a. syntax
   b. a computer language
   c. a program

6. A BASIC instruction begins with a
   a. line number
   b. print statement
   c. quotation mark

7. A word or group of characters is called a
   a. statement
   b. variable
   c. string

8. A question mark may be used for the keyword
   a. INPUT
   b. PRINT
   c. LEN

9. Two statements can be on one line if separated by a
   a. colon
   b. period
   c. semicolon

10. A set of instructions a program branches to is a
    a. subscript
    b. data bank
    c. subroutine

COMPUTER QUIZ #4

Directions: If a statement is true, write T in the blank. If it is false, write F in the blank.

_____ 1. Each time the computer screen is cleared, the program is erased from memory.

_____ 2. An infinite loop is a set of instructions that repeats continuously.

_____ 3. RND stands for random.

_____ 4. A mistake in a computer program is called a bug.

_____ 5. ROM refers to the computer's permanent memory.

_____ 6. CATALOG is a command to file programs alphabetically.

_____ 7. A chip is a path through which electricity flows.

_____ 8. Numbers, letters, and keyboard symbols are used in text displays.

_____ 9. The disk drive contains the computer's memory.

_____ 10. Computers read programs in random order.

_____ 11. Computer programs are called software.

_____ 12. Programs can be erased from RAM by typing NEW.

_____ 13. A pixel is a silicon chip.

_____ 14. GOTO is a branching command.

_____ 15. When a computer is cut off, the program in memory is lost.

_____ 16. A READ statement must be used with a DATA statement.

_____ 17. An exclamation point is the prompt for an INPUT statement.

_____ 18. Flowcharts illustrate decision making of computers.

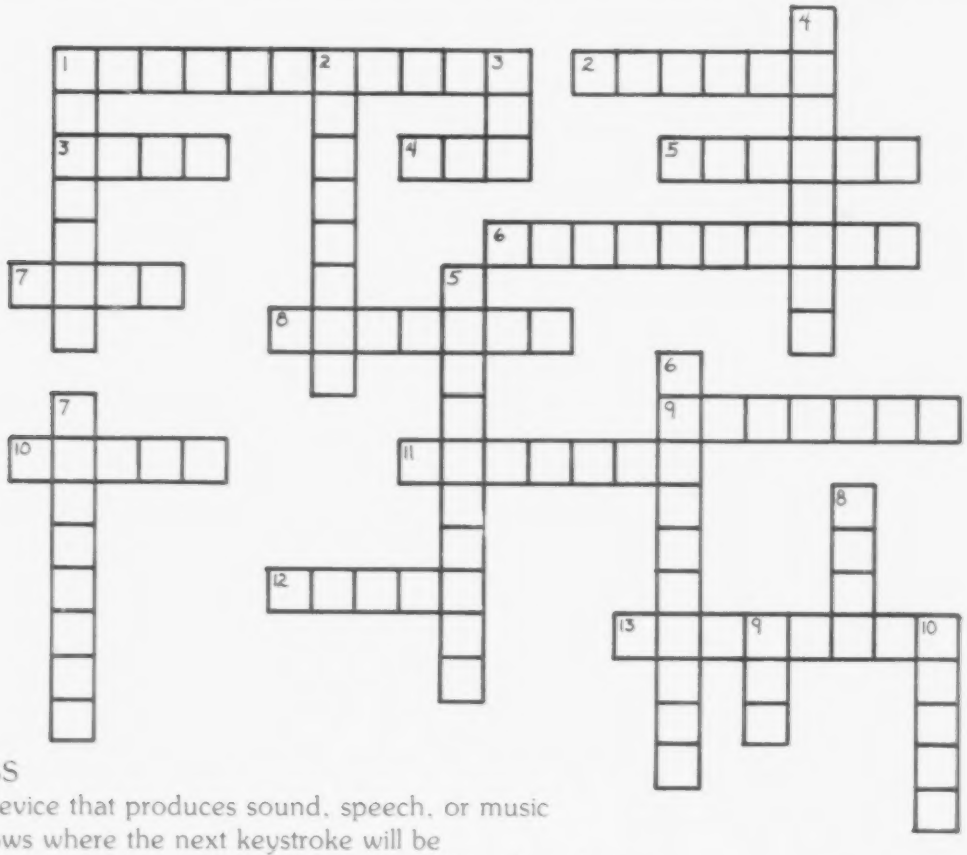
_____ 19. LEN is used in spacing words and phrases.

_____ 20. FLASH is discontinued with the INVERSE statement.

Directions: Can you unscramble the word below? It tells the kind of memory RAM contains.

R E O P R T A M Y

COMPUTER LITERACY CROSSWORD



ACROSS

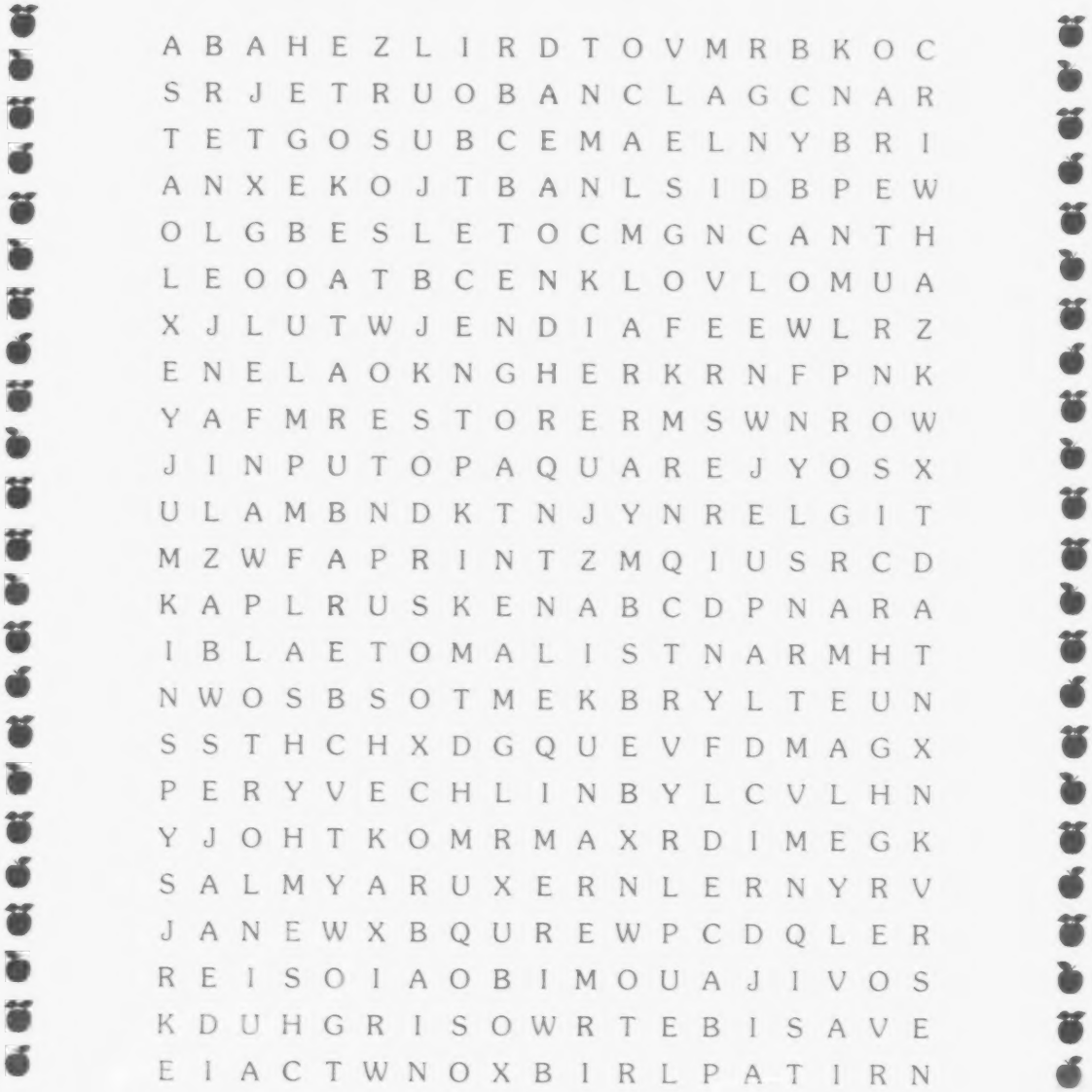
1. a device that produces sound, speech, or music
2. shows where the next keystroke will be
3. a computer language that uses a "turtle" to draw designs
4. random access memory
5. movement of information out of a computer
6. to prepare a disk to receive information
7. to get a computer ready to receive information
8. a set of instructions for the computer
9. to run a program
10. the smallest point of light on a graphics screen
11. a device that gives a paper copy of a program
12. a device to transmit information between computers over telephone wires
13. the part of the computer that resembles a typewriter

DOWN

1. the main ingredient in a computer chip
2. programs used by a computer
3. read only memory
4. drawings shown on a computer screen
5. one who writes instructions for a computer
6. additional equipment attached to expand a computer system
7. a unit of measure to tell how much information a computer can store
8. information
9. a mistake in a program
10. to remove a mistake from a program

BASIC WORD SEARCH

Directions: Thirty BASIC programming terms are hidden in the puzzle below. You may look horizontally, vertically, and diagonally.



A B A H E Z L I R D T O V M R B K O C
S R J E T R U O B A N C L A G C N A R
T E T G O S U B C E M A E L N Y B R I
A N X E K O J T B A N L S I D B P E W
O L G B E S L E T O C M G N C A N T H
L E O O A T B C E N K L O V L O M U A
X J L U T W J E N D I A F E E W L R Z
E N E L A O K N G H E R K R N F P N K
Y A F M R E S T O R E R M S W N R O W
J I N P U T O P A Q U A R E J Y O S X
U L A M B N D K T N J Y N R E L G I T
M Z W F A P R I N T Z M Q I U S R C D
K A P L R U S K E N A B C D P N A R A
I B L A E T O M A L I S T N A R M H T
N W O S B S O T M E K B R Y L T E U N
S S T H C H X D G Q U E V F D M A G X
P E R Y V E C H L I N B Y L C V L H N
Y J O H T K O M R M A X R D I M E G K
S A L M Y A R U X E R N L E R N Y R V
J A N E W X B Q U R E W P C D Q L E R
R E I S O I A O B I M O U A J I V O S
K D U H G R I S O W R T E B I S A V E
E I A C T W N O X B I R L P A T I R N

WORD BANK

LET
RUN
HOME
PLOT
LEN
DATA
RETURN
FLASH
GOTO
INPUT
VLIN
LIST
GR
CLEAR
DIM
INVERSE
END
PRINT
NEW
HLIN
HGR
SAVE
RESTORE
ARRAY
PROGRAM
TAB
REM
READ
GOSUB
TEXT

84

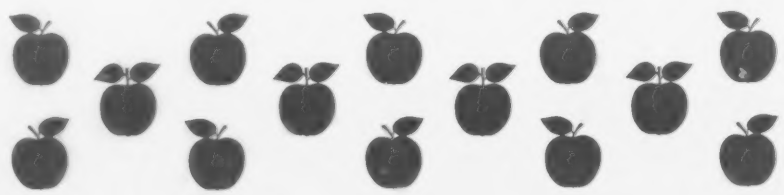
BASIC SCRAMBLER

Directions: Read each clue and unscramble the computer term that is being defined.

1. r m t p o p — a symbol on the screen to show the computer is waiting for a response
   _____
2. r u c r s o — a symbol to show where the next typed character will appear
   _____
3. s g i n t r — a variable consisting of a letter of the alphabet followed by $
   _____
4. a a w e r r d h — computer machinery
   _____
5. n u i t p — information entered into a computer
   _____
6. y a n b r i — a number system using the digits Ø and 1
   _____
7. r s w p o a s d — a code entered to access a program
   _____
8. i o s n l i c — an element computer chips are made from
   _____

Directions: Below are four HINK PINKS. Read the clue for each and see if you can discover the rhyming answers.

Clues:	Words:	
1. fast floppy	_____	_____
2. entertaining command	_____	_____
3. thin remark	_____	_____
4. curved writing statement	_____	_____



MYSTERY ERRORS

Directions: Try to discover the kinds of errors the computer displayed by following the clues.

ERROR #1

1. The first letter is in HURT but not in HUGS.
2. The second letter is in ABLE but not in SELL.
3. The third letter is in NOTE but not in TEXT.
4. The fourth letter is in EDGE but not in DISK.
5. The fifth letter is in EAST but not in NORTH.

What kind of error did the computer display? _____

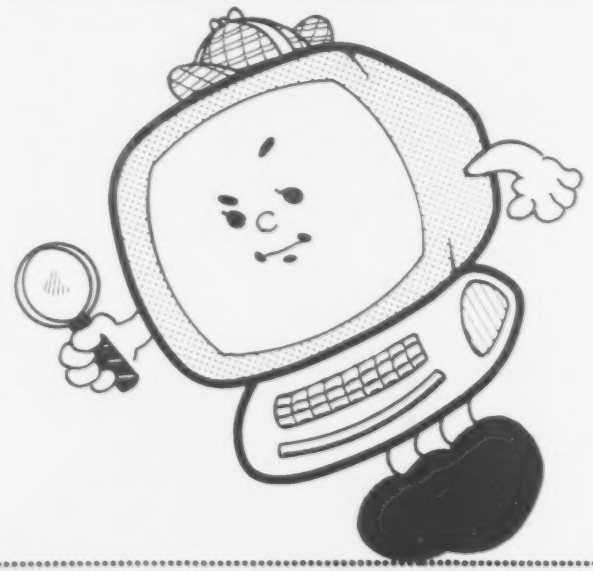
_____

ERROR #2

1. The first letter is in ROSE but not in FLOWER.
2. The second letter is in HAPPY but not in PARTS.
3. The third letter is in NOTES but not in SOLD.
4. The fourth letter is in HABIT but not in BEAR.
5. The fifth letter is in ANSWER but not in QUESTION.
6. The sixth letter is in EXTRA but not in RAINY.

What kind of error did the computer display? _____

_____



ADVENTURE PROGRAMS

Adventure programs are written with three BASIC types of statements: (1) PRINT statements communicate information on the screen. (2) INPUT statements request feedback from the user. (3) IF GOTO statements direct the computer down different paths. Here is a simple example of how an adventure program can be developed.

```
 10 HOME
 20 PRINT "YOU ARE IN A CASTLE."
 30 PRINT "THERE IS A LARGE DOOR IN FRONT OF YOU."
 40 PRINT "DO YOU OPEN IT?"
 50 INPUT A$
 60 IF A$ = "YES" THEN 100
 70 IF A$ = "NO" THEN 200
 80 GOTO 40
100 PRINT "YOU ENTER A SMALL ROOM. YOU SEE AN ENVELOPE ON
    THE TABLE."
110 PRINT "DO YOU OPEN IT?"
120 INPUT B$
130 IF B$ = "YES" THEN 300
140 IF B$ = "NO" THEN 400
150 GOTO 110
200 PRINT "YOU HEAR A STRANGE SOUND."
210 PRINT "DO YOU RUN?"
220 INPUT C$
230 IF C$ = "YES" THEN 500
240 IF C$ = "NO" THEN 600
250 GOTO 210
300 PRINT "THE ENVELOPE CONTAINS A TREASURE MAP. YOU FOLLOW
    IT, LOCATE THE TREASURE,"
310 PRINT "AND CLAIM THE RICHES!"
320 END
400 PRINT "YOU WILL NEVER KNOW WHAT TREASURE YOU OVER-
    LOOKED."
410 END
500 PRINT "YOU GOT AWAY JUST IN TIME! THE ROOF WAS FALLING IN!"
510 END
600 PRINT "THE ROOF IS FALLING!"
610 PRINT "YOU'D BETTER RUN NOW!"
620 PRINT
630 PRINT "YOU BARELY MADE IT!"
640 END
```

COMPUTER WORD LINKS

Directions: Combine syllables below to form a word to match each definition. Write each word beside its definition.

_____ 1. a set of instructions for a computer

_____ 2. a machine that accepts, stores, processes, and gives information

_____ 3. a type of programming error

_____ 4. to appear to create movement

_____ 5. an input device for typing data

_____ 6. a tiny dot of light in graphics

_____ 7. to correct a program error

_____ 8. a computer phone connection

_____ 9. computer programs

_____ 10. the video screen

_____ 11. part of the computer that stores data

_____ 12. information entered into a computer

_____ 13. information received from a computer

_____ 14. symbol showing where the next typed character will appear on the screen

COMPUTER LITERACY WORD SEARCH

Directions: Can you find 32 words related to computers? You may look horizontally, vertically, and diagonally.

C A R Z T O S B H G H L S O N E T R K O M T
A O B J P B K O D A T A M Z P U R S T A E N
P E M S A R C J L E R M T R B D O R S E L T
C H I P O L A S T A L S O F T W A R E K I N
J O L E U I S T E R P O M O N I T O R Y S P
M I C L T T S O T N A Y R Y O S D B O L T E
O D S A P T E R M I N A L K P E B N E Y I R
D A B E U G T R H J E N O M R S A P C I F I
E R S U T V T A D W Z B D L O J S T O N E P
M G K R O N E T W O R K I N G S I E V T W H
T A T E S O I B F G O R S O R N C F M E T E
M B M L O H T F S E Y L K C A O E B T R A R
X K E Y B O A R D S C E R I M K L U J F K A
Y A M L E R U R X V B E N R D A P E S A O L
K E O N U G I O D O U A J B O N S A T C R C
I A R L M A J M S W A N J N I A R S L E T A
L N Y B O R S A T E A J G A R T L M O L N Z
O T P R I N T E R R D R X R B P I B E C B O
B E A C X D E C I B Q V E F A C H X A L H I
Y S H N A K S E G Q U E M D E P I L O T A E
T E W F H A Y A D E B Y T E T P H G S N D V
E L F O B J X R A J C G D H S E D I E A L S
N A D S O C Z L O G O F T A T R N S C K O T
R H A C K E R B H B S X L G U J V H B S V M
S O N E L B U G H R I T W V T S G R A M B H

WORD BANK

DISK	CASSETTE	PROGRAM	INPUT
COMPUTER	TERMINAL	KEYBOARD	CHIP
MODEM	HARDWARE	SOFTWARE	OUTPUT
MONITOR	PRINTER	MEMORY	BYTE
BIT	KILOBYTE	INTERFACE	DATA
GRAPHICS	BASIC	LOGO	PILOT
NETWORKING	PIXEL	PERIPHERAL	HACKER
LIST	ROM	RAM	BUG

BASIC COMMANDS

RUN—carries out a program.

LIST—displays a list of program instructions.

NEW—clears the computer's memory for a different program.

SAVE—writes a program to a disk or cassette tape.

LOAD—reads a copy of the program from the disk or tape into the computer's memory.

COMMANDS DO NOT USE LINE NUMBERS. STATEMENTS WITH KEYWORDS MUST HAVE LINE NUMBERS.

BASIC KEYWORDS

PRINT—tells the computer to display whatever is inside quotation marks. When used alone, PRINT instructs the computer to display a blank line.

LET—assigns a value to a variable.

INPUT—requests information from the user.

GOTO—sends the computer to a different line number.

IF THEN—sends the computer to the given line if the condition is true.

STOP—brings the program to a halt.

FOR . . . (STEP)—runs a loop a specified number of times.

NEXT—closes a loop.

TAB—spaces the print at a specified position in a line.

READ—calls for information stored in DATA statements.

DATA—holds information called for with a READ statement.

GLOSSARY

address—a location in a computer's memory.

BASIC—Beginner's All-purpose Symbolic Instruction Code. It is a programming language that is easy to use.

bug—a mistake in a computer program.

catalog—a directory or list of files on a list.

CATALOG—a DOS command used to list the contents of a disk.

character—any written symbol such as a letter, digit, or punctuation mark.

COLOR—used to select a color in low resolution graphics. (Example: COLOR = 1)

command—information typed on a computer keyboard telling the computer to perform a certain action. RUN is an example of a command that directs the computer to carry out a program.

computer—an electronic machine that stores, transfers, and changes programmed information as a programmer instructs.

computer system—a computer, its hardware, software, and firmware.

concatenate—to join two or more strings together into a single string.

cursor—the screen marker that indicates where the next character is to be displayed when it is typed.

DATA—information to be used with the READ statement. DATA instructions are useful when there are long lists of numbers or spelling words.

debug—a term meaning to correct a mistake in a computer program.

diskette—a small (5¼ inch) flexible disk used to save programs. It is used with a disk drive and a computer.

END—an optional BASIC statement at the end of a program. It is used to denote the conclusion of a program.

FLASH—a statement that causes text to flash on the screen. It is discontinued with the NORMAL statement.

GOTO—a keyword used to direct the computer to a specific program line.

GR—a statement that tells the computer to display low-resolution graphics. GR stands for graphics.

HGR—a statement that tells the computer to display page 1 of high-resolution graphics. HGR means high-resolution graphics. HGR2 refers to page 2 of high-resolution graphics.

HLIN—a statement that is used to draw horizontal lines.

HOME—a statement that clears the text from the screen and places the cursor at the top left corner of the screen.

HTAB—a statement that moves the cursor to a specified horizontal position.

input—information that enters a computer.

INPUT—a keyword used to request information from the keyboard.

INVERSE—a statement that causes the PRINT statements that are used after it to be displayed in black on a white background.

keyboard—the part of the computer that has keys like a typewriter and is used to type information to the computer.

keyword—a special word in BASIC that helps the computer identify its instructions. PRINT is an example of a keyword.

LEN—a function that counts the number of characters in a string.

line number—the number that is typed at the beginning of a statement. It tells the computer in what order to carry out instructions.

LIST—a command used to direct the computer to display the line by line instructions of a program.

loop—part of a program that is repeated continually until it is stopped by the user or by an instruction that directs it to repeat a specific number of times.

memory—the part of the computer that stores information.

NEXT—used to mark the end of a loop and cause the loop to be repeated.

NEW—a command typed to clear the computer's memory. It is used when the operator wants to begin a new program and needs the old one out of memory.

NORMAL—a statement used to cancel FLASH or INVERSE. It causes the text to appear on the usual white-on-black screen.

output—information coming from a computer. It may be on the screen or printed on paper.

PRINT—a keyword used to tell the computer to print a program statement. The words that are to be printed are enclosed inside quotation marks. If no words are used after PRINT, a blank line is displayed.

program—a set of instructions to be carried out by a computer.

random (RND)—a function used to produce numbers between Ø and 1 as if they were drawn from a hat. When used with the integer function (INT), random whole numbers are produced. RND means random.

READ—a statement that gets information from a DATA statement within a program. See DATA.

REM—means remark. The computer does not print a program line that begins with REM. REM makes clearer documentation in a listing.

RUN—a command to execute a program.

SAVE—a command used with the name of a program to store it to a disk or tape.

statement—an instruction to the computer. It is part of a program and follows a line number.

string—any character or set of characters inside quotation marks. $ is the symbol for string.

syntax—the rules concerning the program language structure.

syntax error—a message to the user that a mistake has been made in spelling, punctuation, or some other language structure of a program.

TAB—used with PRINT to move the cursor to a specified horizontal position.

text—numbers, letters, and other keyboard symbols.

TEXT—the command used to change the screen from graphics to text.

variable—(1) a place in the memory of a computer where a specific value can be stored; (2) a symbol used in a program to represent a place where a specific value is stored. Example: C = 5. The variable C contains a value of 5.

VLIN—a statement instructing the computer to draw a vertical line.

VTAB—used to move the cursor to a specified vertical position.

ANSWER KEY

Answers are not given for all pages since interaction with the computer provides the needed results. However, answers are provided to quizzes and pages containing important concepts.

PAGE 8

1. Quotation marks are needed to tell the computer exactly what to print.
2. A syntax error will result if quotation marks are not used in a statement that is to be printed.

PAGE 34

1. 10Ø
2. 11Ø
3. The program goes to Line 13Ø and prints the message inside the quotation marks.

PAGE 38

1. The first PRINT in Line 5Ø tells the computer to print a blank line. The second PRINT tells the computer to print the message that is enclosed inside quotation marks.
2. RND is a random function that works like selecting numbers out of a hat.
3. REM in Line 1Ø means remark. It is a message to the programmer but is not necessary for execution of the program.
4. Line 10Ø is not reached because the GOTO creates a loop in the program.

PAGE 41

1. A semicolon tells the computer to space the information together with no spaces between words or numbers.
2. A comma tells the computer to space the information apart. The output will be in columns.

PAGE 51

1. 3Ø 2. 4Ø 3. 8Ø AND 10Ø 4. 12Ø
5. Line 1Ø puts the output at the top left corner of the screen.

PAGE 70

Correct: 3, 5, 8, 13, 15, 16, 18, 19, 20

PAGE 73

1. NEW, RETURN
2. LIST, RETURN
3. RUN, RETURN
4. ?, PRINT

1. K 2. C 3. K 4. K 5. K 6. C 7. K 8. C 9. K 10. K

1. D 2. C 3. E 4. A 5. F 6. G 7. H 8. B

PAGE 74

1. 5 2. 2 + 3 3. HELLO 4. ? 5. If there was a previous Line 10, it will be deleted. Nothing will be displayed on the screen. 6. GOOD MORNING 7. 12 8. 4 9. Nothing is displayed on the screen. 10. THE SUM OF 2 AND 6

PAGE 75

1. b 2. b 3. a 4. a 5. c 6. a 7. c 8. b 9. a 10. c

PAGE 76

1. F 2. T 3. T 4. T 5. T 6. F 7. F 8. T 9. F 10. F 11. T 12. T 13. F 14. F 15. T 16. T 17. F 18. T 19. F 20. F

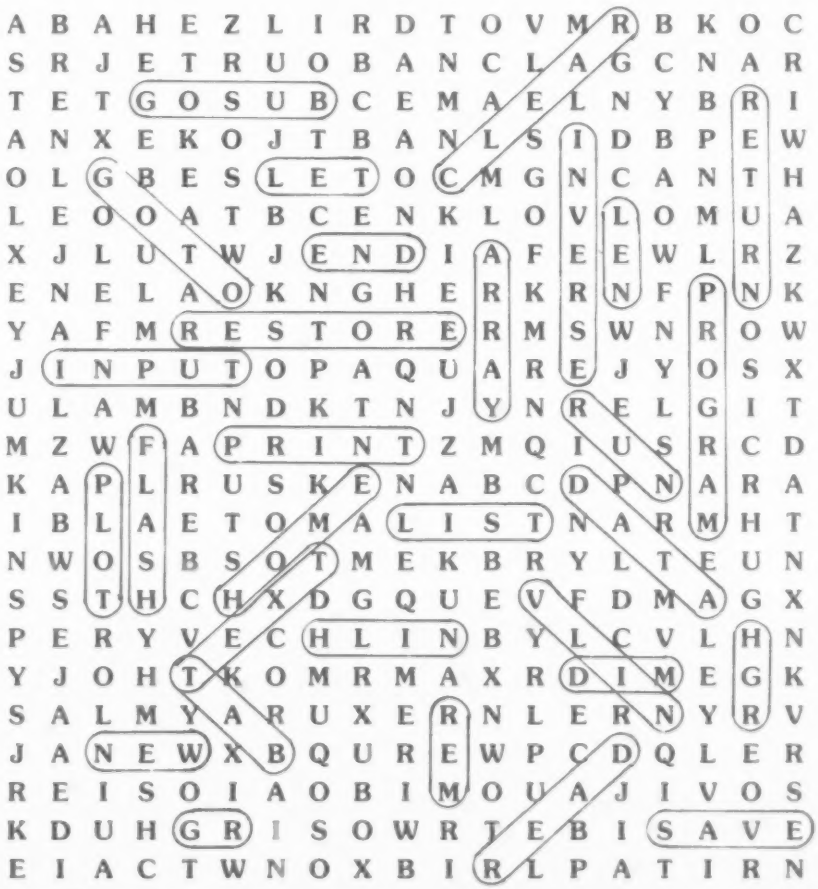
TEMPORARY

PAGE 77

ACROSS: 1. SYNTHESIZER 2. CURSOR 3. LOGO 4. RAM 5. OUTPUT 6. INITIALIZE 7. BOOT 8. PROGRAM 9. EXECUTE 10. PIXEL 11. PRINTER 12. MODEM 13. KEYBOARD

DOWN: 1. SILICON 2. SOFTWARE 3. ROM 4. GRAPHICS 5. PROGRAMMER 6. PERIPHERAL 7. KILOBYTE 8. DATA 9. BUG 10. DEBUG

PAGE 78


A B A H E Z L I R D T O V M R B K O C
S R J E T R U O B A N C L A G C N A R
T E T G O S U B C E M A E L N Y B R I
A N X E K O J T B A N L S I D B P E W
O L G B E S L E T O C M G N C A N T H
L E O O A T B C E N K L O V L O M U A
X J L U T W J E N D I A F E E W L R Z
E N E L A O K N G H E R K R N F P N K
Y A F M R E S T O R E R M S W N R O W
J I N P U T O P A Q U A R E J Y O S X
U L A M B N D K T N J Y N R E L G I T
M Z W F A P R I N T Z M Q I U S R C D
K A P L R U S K E N A B C D P N A R A
I B L A E T O M A L I S T N A R M H T
N W O S B S O T M E K B R Y L T E U N
S S T H C H X D G Q U E V F D M A G X
P E R Y V E C H L I N B Y L C V L H N
Y J O H T K O M R M A X R D I M E G K
S A L M Y A R U X E R N L E R N Y R V
J A N E W X B Q U R E W P C D Q L E R
R E I S O I A O B I M O U A J I V O S
K D U H G R I S O W R T E B I S A V E
E I A C T W N O X B I R L P A T I R N

PAGE 79

1. PROMPT 2. CURSOR 3. STRING 4. HARDWARE 5. INPUT 6. BINARY
7. PASSWORD 8. SILICON 1.BRISK DISK 2. FUN RUN 3. SLIM REM
4. BENT PRINT

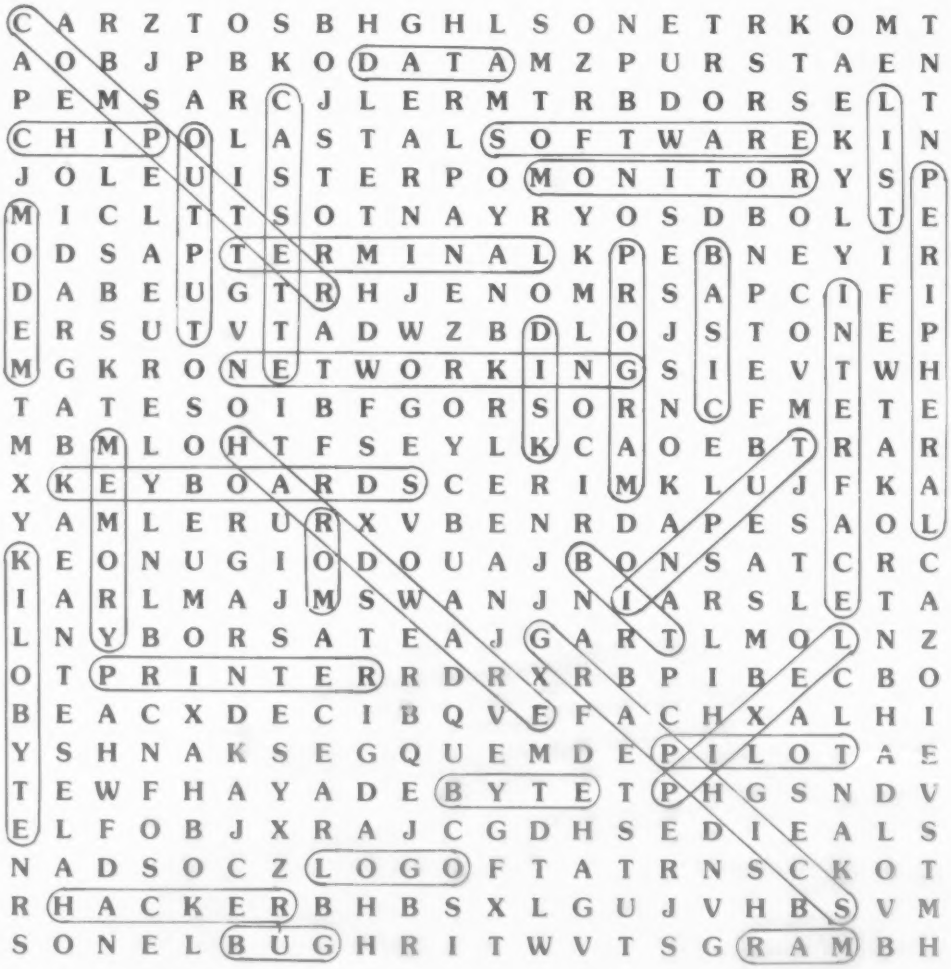
PAGE 80

1. RANGE 2. SYNTAX

PAGE 82

1. PROGRAM 2. COMPUTER 3. SYNTAX 4. ANIMATE 5. KEYBOARD
6. PIXEL 7. DEBUG 8. MODEM 9. SOFTWARE 10. MONITOR 11. MEMORY
12. INPUT 13. OUTPUT 14. CURSOR

PAGE 83


C A R Z T O S B H G H L S O N E T R K O M T
A O B J P B K O D A T A M Z P U R S T A E N
P E M S A R C J L E R M T R B D O R S E L T
C H I P O L A S T A L S O F T W A R E K I N
J O L E U I S T E R P O M O N I T O R Y S P
M I C L T T S O T N A Y R Y O S D B O L T E
O D S A P T E R M I N A L K P E B N E Y I R
D A B E U G T R H J E N O M R S A P C I F I
E R S U T V T A D W Z B D L O J S T O N E P
M G K R O N E T W O R K I N G S I E V T W H
T A T E S O I B F G O R S O R N C F M E T E
M B M L O H T F S E Y L K C A O E B T R A R
X K E Y B O A R D S C E R I M K L U J F K A
Y A M L E R U R X V B E N R D A P E S A O L
K E O N U G I O D O U A J B O N S A T C R C
I A R L M A J M S W A N J N I A R S L E T A
L N Y B O R S A T E A J G A R T L M O L N Z
O T P R I N T E R R D R X R B P I B E C B O
B E A C X D E C I B Q V E F A C H X A L H I
Y S H N A K S E G Q U E M D E P I L O T A E
T E W F H A Y A D E B Y T E T P H G S N D V
E L F O B J X R A J C G D H S E D I E A L S
N A D S O C Z L O G O F T A T R N S C K O T
R H A C K E R B H B S X L G U J V H B S V M
S O N E L B U G H R I T W V T S G R A M B H

INDEX

ADDITION 28,44,45,49,74
ADDRESS 69,85
ARITHMETIC 44-46,49-51
AVERAGING 27,51
BASIC iv,1-3,12,30,71,72,78,84,85
BINGO 54,55
BIT 69
BUG 85
CATALOG 11,85
CHARACTER 26,39,85
COLOR 61,62,66,85
COLON 7,75
COMMA 41,44
COMMAND 11,73,75,84,85
CONCATENATE 28,29,85
CONTROL-C 12,15,33
CONTROL-S 12
COUNTING 34,46
CURSOR 34,79,82,85
DATA 35,36,77,83,85
DIVISION 44,45,50
DELETE 16
EDITING 14-16
END 1,2,10,72,73,85
FLAG 36
FLASH 59,76,85
FLOWCHART 32,33,76
FOR . . . NEXT 10,34,46,55,84
GOTO 10,17,18,30,33,47-50,55,73,76,84,85
GR 61,62,86
GRAPHICS 60-69
GREATER THAN SYMBOL (▶) 40,47,48
GRID 60,61,63,64
HCOLOR 65-68
HGR 65-69,86
HGR2 65,69
HIGH-RESOLUTION 65-69
HLIN 62,63,78,86
HOME 34,78,86
HPLOT 67,68
HTAB 43,59,86
IF . . . THEN 10,30-33,52,81,84
INFINITE LOOP 17,18,73,76,86
INPUT 5,7,10,21,22,27,29,33,40,48-52,59,74,84,86
INT 37,38,50

INVERSE 58,86
KEYBOARD 1,9,39,57,86
KEYWORDS 10,35,73,84,86
LEN 39,40,52,76,86
LESS THAN SYMBOL(◄) 47,48
LET 10,21,25-27,51,73
LINE NUMBER 1-3,8,11,19,86
LIST 3,4,6,11,12,16,71,73,84,86
LOAD 11,84
LOOP 17,18,73,76,86
LOW-RESOLUTION 61-64
MATHEMATICS 44-47,49,50
MEMORY 23,24,71,86
MULTIPLE STATEMENTS 7
MULTIPLICATION 44,45,49
NEXT 10,34,66,69,73,86
NEW 11,12,21,34,35,71,73,78,84,86
NORMAL 58,59,86
OUTPUT 28,41,82,87
PIXEL 66,67,76
PLOT 61-63
PRINT 1,2,6,10,20,74,75,87
PROGRAM 1-3,6,8,9,49,53,73,74,87
QUOTATION MARKS 1,8,19,20,25,35,57
QUIZZES 73-76
RAM 12,75,77
RANDOM 37,38,55,76,87
READ AND DATA 35,36,76,84,87
REM 10,27,52,53,71,73,87
RND 37,38,55,76,87
ROM 76,77,83
RUN 1-3,5,71,75,84,87
SAVE 11,84,87
SEMICOLON 41-44
SPACING 25,26,41-43,50
STATEMENT 2,3,5,7,10,18-20,87
STOP 10,84
STRING 22,26,28,29,35,36,73,87
SUBTRACTION 44,45,49
SYNTAX ERROR 2,13,14,19,80,87
TAB 42,87
TEXT 60,61,78,87
VARIABLE 23-26,34,39,87
VLIN 62,63,87
VTAB 43,59,87

THE GROWING FAMILY OF GOOD APPLE PRODUCTS INCLUDES:

PERIODICALS . . . five to choose from . . . each will bring you a wealth of ideas.

★ **THE GOOD APPLE NEWSPAPER** . . . Ideas! Ideas! Ideas! That is what *The Good Apple Newspaper* is all about. Each issue, and there are five each school year, is a treasure-house for the 2-8 teacher. Games, posters, units of learning, task cards, bulletin board ideas, etc. Ideal for all subject areas, adaptable for all grade levels . . . innovative, skill-building, easy-to-use ideas for you and your students. (For 2-8.)

★ **LOLLIPOPS, LADYBUGS AND LUCKY STARS** . . . And you will thank your lucky stars for finding this newspaper of ideas and activities for teachers of preschool and primary children. Each year there are five 32-page issues chock-full of delightful learning experiences for early learners: patterns, poems, stories, readiness skills, reinforcement activities for all subject areas, arts and crafts, reading, music! (For Preschool-2.)

★ **CHALLENGE . . . reaching & teaching the GIFTED child** . . . for parents and teachers of gifted, talented and creative children of all ages. Our **CHALLENGE** at Good Apple is to put the world of gifted education at your fingertips—identification, evaluation, methods of instruction, materials, model programs, units of study, mind benders—with the nation's prominent authorities in the field helping. Joe Wayman, editor. (For K-8.)

★ **SHINING STAR** . . . an exciting Christian education publication. Over 80 pages of practical teaching techniques and suggestions for educators teaching in private schools, church programs and parents seeking material for home use. Each issue will include approximately 40 reproducible work sheets, plus page after page of Bible quiz activities, family worship, bulletin board ideas, music-related fun, arts and crafts, gameboards, and Biblical heroes. (For K-8.)

★ **EDUCATIONAL OASIS** . . . adds enthusiasm and zest to the middle grade classroom through 64 pages of motivating ideas and activities. Each issue features reproducible activity sheets, bulletin board ideas, current topical articles, reviews, calendars, a 16" x 12" full-color poster with accompanying activities and much more. Five issues each school year. (For 5-9.)

BOOKS . . . over 350 titles in all subject areas for all grade levels, preschool—12. Idea books, activity books, bulletin board books, units of instruction all with many reproducible pages.

READING	READINESS	VOCABULARY	SPELLING
BOOK REPORTS	DICTIONARY FUN	LANGUAGE ARTS	CREATIVE WRITING
CREATIVITY	GIFTED EDUCATION	BASIC SKILLS	STORYTELLING
GAMEBOARD IDEAS	POETRY	MATH (all areas)	METRICS
LEARNING CENTERS	CONTRACTS	OUTDOOR EDUCATION	INDEPENDENT STUDY
SCIENCE	SOCIAL STUDIES	SELF-CONCEPT	MEDIA—TELEVISION
ARTS/CRAFTS	PRESCHOOL	PRIMARY	MIDDLE SCHOOL
HOLIDAY IDEAS	BULLETIN BOARDS	TEACHER HELPERS	WORD GAMES
CAREER EDUCATION	PHYSICAL EDUCATION	HEALTH	AWARDS
MOTIVATION	MUSIC	ART EDUCATION	CLASSROOM ENVIRONMENT

GOOD APPLE MIGHT HAVE JUST THE VERY BOOK YOU HAVE BEEN LOOKING FOR!

★ **RECORDS—CASSETTES** . . . a dozen favorites with accompanying activity books.
★ **ACTIVITY POSTERS** . . . over 30 titles. Your students will love to complete them.
★ **NOTE PADS** . . . Several choices to reward and motivate. A note for almost every occasion.

WORKSHOPS . . . Good Apple will provide you with a unique variety of workshops, from gifted education to positive self-concept, from make-it and take-it to major content areas of curriculum. Twenty outstanding consultants are available to meet your needs. All of the Good Apple professionals are experienced teachers and recognized authorities. Our programs include scheduled workshops, special-purpose workshops designed specifically for your school and summer institutes where YOU will be the focus of a 3-5 day involvement experience and have the opportunity for graduate credit. We will work together on providing a workshop to fit your needs! Write to: Good Apple Workshops, Box 299, Carthage, IL 62321-0299 or call Polly Aten at 217-357-3981.

FOR A FREE CATALOG WRITE: Good Apple, Inc., Box 299, Carthage, IL 62321-0299

GOOD APPLE, INC.
BOX 299
CARTHAGE, IL 62321